

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA TRANSPORTE E LOGÍSTICA

JÚLIA CAROLINE NUNES

MODELAGEM MATEMÁTICA PARA OTIMIZAÇÃO DE ESCALONAMENTO DE
TAREFAS ENTRE COLABORADORES

Joinville

2018

JÚLIA CAROLINE NUNES

MODELAGEM MATEMÁTICA PARA OTIMIZAÇÃO DE ESCALONAMENTO DE
TAREFAS ENTRE COLABORADORES

Trabalho apresentado como requisito para
obtenção do título de bacharel no Curso de
Graduação em Engenharia de Transporte e
Logística do Centro Tecnológico de Joinville
da Universidade Federal de Santa Catarina.

Orientadora: Dra. Silvia Lopes de Sena
Taglialenha.

Joinville

2018

JÚLIA CAROLINE NUNES

MODELAGEM MATEMÁTICA PARA OTIMIZAÇÃO DE ESCALONAMENTO DE
TAREFAS ENTRE COLABORADORES

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de bacharel em Engenharia de Transporte e Logística, na Universidade Federal de Santa Catarina, Centro Tecnológico de Joinville.

Banca Examinadora:

Dra. Silvia Lopes de Sena Taglialenha
Orientadora
Presidente

Dr. Romulo Alberto Castillo Cardenas
Membro
Universidade Federal de Santa Catarina

Dr. Cristiano Vasconcellos Ferreira
Membro
Universidade Federal de Santa Catarina

AGRADECIMENTOS

Em primeiro lugar, a Deus, pela graça da vida e por todas as oportunidades dadas a mim durante o decorrer dela.

À Prof.^a Dra. Silvia Taglialenha, orientadora deste trabalho, por todo o seu empenho, não só científico, mas também humano, pela paciência, motivação, disponibilidade e dedicação que teve para comigo. Foi de fato uma enorme honra poder desenvolver este trabalho sob a sua coordenação e orientação.

Aos meus pais, Ricardo e Taísa, por terem me ensinado a viver, indo de encontro aos meus objetivos através do meu esforço. Por realizarem todas aquelas pequenas coisas que me permitiam estar dedicada apenas ao trabalho. Se conquistei mais essa vitória, devo a vocês.

Ao meu namorado, Lucas Galon, por estar sempre disponível para me escutar, me compreender e por encontrar sempre motivação para seguir em frente.

Aos meus amigos, por todo o incentivo e apoio, em especial a Ana Goerz, pelo seu esforço em atender as exigências profissionais na minha ausência.

À Associação Onda Dura, que além de ter sido o estudo de caso avaliado neste trabalho, foi a equipe de trabalho, e família, que me deu todo o suporte necessário para concluir essa etapa da minha história.

*“O sucesso não é medido pelo nível de dificuldade do problema com o qual se lida, mas sim por sua reincidência”
John Foster Dulles.*

RESUMO

Este trabalho tem como proposta mostrar a aplicabilidade de um modelo matemático em um estudo de caso real, para escalonamento de tarefas entre colaboradores que trabalharam em um evento, utilizando um método exato para resolução do problema *flow shop*. Este problema é classificado na literatura como NP-difícil e consiste em determinar o sequenciamento de tarefas de um projeto, objetivando a minimização do *makespan* no projeto, o tempo total do processo, assim como gerenciar os recursos disponíveis de forma otimizada. Tudo isso considerando as restrições de precedência entre as tarefas, fornecendo assim, uma ordem dos serviços a serem executados. Para tanto, baseou-se no modelo de programação linear inteira mista proposto em “Otimização do sequenciamento de tarefas em máquinas paralelas com tempos de processamento diferentes” (BARBOSA, 2017) para obter a resolução do problema. Apresenta-se uma forma de resolução exata do modelo por meio de *solvers* processados remotamente. Para a adaptação do modelo definiu-se as restrições considerando as particularidades do projeto e preceitos estabelecidos de forma a otimizar o escalonamento. Considerou-se sua aplicação em cenários estabelecidos com solução ótima conhecida previamente, iniciando os testes com número de entrada de dados de tarefas e colaboradores reduzidos, expandindo ao número máximo de processamento que o *solver* admitiu. Os resultados obtidos com a aplicação dos dados reais aproximados da organização, demonstraram a possibilidade de melhorar a utilização dos recursos humanos disponíveis, assim como reduzir a quantidade de colaboradores envolvidos nas tarefas, reduzindo por consequência os custos envolvidos nas equipes de trabalho. Essa redução é economicamente considerável, uma vez que além de aumentar os lucros da organização, também possibilita uma melhora no nível do serviço entregue ao cliente. Para fins de comparação dos resultados, aplicou-se um modelo alterando a função objetivo para minimizar o número de colaboradores, ao invés da multa por tempo. Os resultados obtidos, atendem as necessidades de resolução do problema e academicamente, contribuem para o aperfeiçoamento do problema em uma modelagem futura.

Palavras-chave: Designação de tarefas. Modelo matemático para escalonamento de tarefas. Scheduling.

ABSTRACT

This paper aims to show the applicability of a mathematical model in a real case study, for scheduling tasks between employees who worked in an event, using an exact method to solve the problem flow shop. This problem is classified in the literature as NP-difficult and consists of determining the task sequencing of a project, aiming at the minimization of the makespan in the project, the total time of the process, as well as to manage the available resources in an optimized way. All this considering the precedence constraints between the tasks, thus providing an order of the services to be executed. For this, it was based on the model of mixed integer linear programming proposed in "Optimization of the sequencing of tasks in parallel machines with different processing times" (BARBOSA, 2017) to obtain the resolution of the problem. An exact form of resolution of the model is presented through solvers processed remotely. For the adaptation of the model the constraints were defined considering the particularities of the project and established precepts in order to optimize the scheduling. Its application was considered in scenarios established with optimal solution known previously, starting the tests with input number of tasks data and reduced employees, expanding to the maximum number of processing that the solver admitted. The results obtained with the application of the actual approximate data of the organization showed the possibility of improving the use of available human resources, as well as reducing the number of employees involved in the tasks, thus reducing the costs involved in the work teams. This reduction is economically considerable, since in addition to increasing the profits of the organization, it also enables an improvement in the level of service delivered to the customer. For purposes of comparison of the results, a model was applied altering the objective function to minimize the number of employees, instead of the time penalty. The results obtained, meet the problem solving needs and academically, contribute to the improvement of the problem in a future modeling.

Keywords: Assignment of tasks. Exact model for scheduling tasks. Scheduling.

LISTA DE FIGURAS

Figura 1 - Base da Gestão de um Projeto	23
Figura 2 - Exemplo de processamento em máquinas em <i>Job Shop</i>	30
Figura 3 - Exemplo de processamento em máquinas em <i>Open Shop</i>	30
Figura 4 - Exemplo de processamento em máquinas em <i>Flow Shop</i>	31
Figura 5 - Escalonamento de tarefas entre colaboradores	33
Figura 6 -Processo de modelagem.....	35
Figura 7 -Registro fotográfico do Evento “Conferência Atos 2018”	41
Figura 8 - Registro fotográfico do Evento “Conferência Atos 2018”	41
Figura 9 - Processo de Decisão para Escalonamento de Tarefa	42
Figura 10 - Modelo completo em AMPL	52
Figura 11 - Modelo completo em AMPL	53
Figura 12 - Modelo sintetizado em AMPL arquivo de dados	54
Figura 13 - Modelo para 4 tarefas e 3 colaboradores em AMPL	56
Figura 14 - Modelo para 4 tarefas e 3 colaboradores em AMPL arquivo de dados.....	57
Figura 15 – Escalonamento entre 3 Colaboradores	57
Figura 16 - Modelo para 12 tarefas e 12 colaboradores em AMPL	59
Figura 17 - Modelo para 12 tarefas e 12 colaboradores em AMPL arquivo de dados	60
Figura 18 - Escalonamento entre 12 colaboradores	61
Figura 19 – Modelo para 33 tarefas e 30 colaboradores em AMPL.....	61
Figura 20 - Escalonamento entre 28 colaboradores	63
Figura 21 - Modelo para 4 tarefas e 3 colaboradores em AMPL	64
Figura 22 - Designação de 4 tarefas	65
Figura 23 - Modelo para 12 tarefas e 12 colaboradores em AMPL	66
Figura 24 - Designação de 12 tarefas	67
Figura 25 - Modelo para 30 tarefas e 28 colaboradores em AMPL	68
Figura 26 – Designação de 30 tarefas.....	69

LISTA DE TABELAS

Tabela 1 - Dados para escalonamento de 4 tarefas entre 3 colaboradores	55
Tabela 2 - Dados para escalonamento de 12 tarefas entre 12 colaboradores	58

LISTA DE ABREVIATURAS E SIGLAS

FSS: *Flow Shop Schedule*

OC: Otimização Combinatória

PI: Programação Inteira

PO: Pesquisa Operacional

PL: Programação Linear

PLI: Programação Linear Inteira

PLIM: Programação Linear Inteira Mista

PCP: Planejamento e controle de produção

PPP: Problema de Programação de Produção

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVOS	16
1.1 Objetivo Geral	16
1.2 Objetivos Específicos.....	17
1.2 JUSTIFICATIVA	17
1.2.1 Acadêmica	18
1.2.2 Financeira.....	18
2 FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 PROGRAMAÇÃO LINEAR INTEIRA	20
2.2 GESTÃO DE PROJETOS.....	22
2.2.1 Programação da Produção	25
2.2.2 Problemas de Programação de Projetos.....	26
2.2.3 Problema de Planejamento (<i>scheduling</i>)	29
2.2.4 Programação em <i>Flow Shop</i> (<i>Flow Shop Schedule</i>)	31
2.3 O PROBLEMA DE ESCALONAMENTO DE TAREFAS	32
2.4 SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS PARALELAS	36
3 MÉTODO DE SOLUÇÃO.....	39
3.1 MODELO DE DESIGNAÇÃO DE TAREFAS PARA O ESTUDO DE CASO	40
3.1.1 Definição das variáveis.....	43
3.1.2 Definição das Restrições.....	44
<u>3.1.2.1 Restrições de não interferência entre as tarefas.....</u>	<u>44</u>
<u>3.1.2.2 Restrição de Cobertura</u>	<u>45</u>
<u>3.1.2.3 Restrição de Empacotamento</u>	<u>45</u>
<u>3.1.2.4 Restrições de Cumprimento do Prazo de Execução</u>	<u>45</u>
<u>3.1.2.5 Restrição de Precedência entre as tarefas</u>	<u>46</u>
<u>3.1.2.6 Multa por atraso.....</u>	<u>47</u>
3.1.3 Modelo Algébrico considerando n tarefas e m colaboradores.....	47
3.2 MODELO PARA OTIMIZAÇÃO DO NÚMERO DE COLABORADORES	48
3.2.1 Definição de Variáveis.....	48
3.2.2 Definição das Restrições.....	48
<u>3.2.2.1 Restrição de garantia do Colaborador executar alguma tarefa</u>	<u>49</u>
<u>3.2.2.2 Restrição para limitar o número de Colaboradores</u>	<u>49</u>

3.2.2.3 Restrição de capacidade de tarefas por colaborador.....	49
3.2.2.4 Quantidade de Colaboradores.....	49
3.2.3 Modelo Algébrico para otimização dos colaboradores	50
4 RESULTADOS COMPUTACIONAIS	51
4.1 DETERMINAÇÃO DAS AMOSTRAS DO ESTUDO DE CASO.....	51
4.2 IMPLEMENTAÇÃO	52
4.2.1 Modelo de busca pela redução da Multa	52
4.2.2 Modelo de busca pela redução de Colaboradores	53
4.2.3 Dados para entrada dos modelos	54
4.3 SOLUÇÃO PROPOSTA PARA REDUÇÃO DA MULTA.....	55
4.3.1 Modelo Algébrico considerando 4 tarefas e 3 colaboradores	55
4.3.2 Modelo Algébrico considerando 12 tarefas e 12 colaboradores	58
4.3.3 Modelo Algébrico considerando 30 tarefas e 28 colaboradores	61
4.4 SOLUÇÃO PROPOSTA PARA REDUÇÃO DE COLABORADORES	64
4.4.1 Modelo Algébrico considerando 4 tarefas e 3 colaboradores	64
4.4.2 Modelo Algébrico considerando 12 tarefas e 12 colaboradores	66
4.4.3 Modelo Algébrico considerando 30 tarefas e 28 colaboradores	68
4.5 ANÁLISE DOS MÉTODOS DE SOLUÇÃO	70
5 CONSIDERAÇÕES FINAIS.....	71
5.1 TRABALHOS FUTUROS	72
REFERÊNCIAS	73
APÊNDICE A – DADOS	77

1 INTRODUÇÃO

Para atingir os objetivos estratégicos de uma organização a aplicação otimizada dos recursos se tornará um aspecto cada vez mais imprescindível aos que buscam sobreviver em um mundo totalmente globalizado e informatizado. É natural que empresas procurem a otimização dos processos na execução de um projeto, a fim de atingir um nível de excelência em suas operações, melhorando assim o serviço oferecido aos seus clientes.

Ao visar esse quadro, é notável a premência por otimizar sistemas que envolvem a busca de uma função-objetivo, planejamento de tarefas (*scheduling*), combinadas a restrições impostas a partir de um conjunto discreto e finito de dados. Onde variadas técnicas de otimização têm sido utilizadas, reduzindo ao máximo os custos de execução das atividades e aumentando o rendimento dos processos planejados em um projeto (BOWERSOX, 2011).

O sucesso de um projeto está totalmente ligado ao bom gerenciamento de seus recursos. Existem vários modelos, técnicas e ferramentas que visam dar suporte a gerência de projetos no que tange a alocação de equipes ou tarefas, controle de pessoal e desenvolvimento de cronogramas, seja para projetos em geral ou em um âmbito específico. Isto leva a uma questão fundamental: “Dado um grupo de colaboradores disponíveis e um conjunto de atividades, como alocar os recursos e como sequenciar as atividades de forma a trazer maiores benefícios à organização?”. (TRICHES, 2015)

Por meio deste questionamento deu-se início a proposta deste trabalho. Onde, dentre as técnicas disponíveis, buscou-se a mais indicada para a resolução de um problema de escalonamento de tarefas entre uma equipe de colaboradores, através de um modelo exato adaptado para o contexto do projeto em questão.

Na missão de gerir um projeto, uma das áreas mais estudadas é a dos problemas de escalonamento das tarefas que o compõem. Escalonar significa designar recursos e tempo de início para a execução de cada tarefa, até que todas tenham sido processadas sob as restrições impostas, com o objetivo de minimizar a duração do projeto, reduzindo os custos ou maximizando as receitas (BLAŻEWICZ, 1996). A aplicação pode ser em diversas áreas, exemplos frequentes deste tipo de problemas são: o planejamento de equipamentos produtivos na indústria, conhecido na terminologia anglo-saxónica como *production scheduling*, o

planejamento de obras na construção civil ou a designação de recursos a um projeto, citando apenas alguns.

De acordo com o PMBOK® Guide (PMI, 2013), guia publicado pelo *Project Management Institute*, um projeto pode ser definido como uma sequência de atividades (ou eventos) com início e fim estabelecidos, executadas segundo uma ordem previamente determinada, objetivando alcançar uma meta temporal pré-estabelecida. Estas atividades possuem duração e precedência pré-determinadas. Além disso, demandam de recursos - escassos e limitados - que realizam as atividades (TCHAO, 2007).

Planejar as etapas de um projeto complexo não é uma atividade fácil, devido ao grande número de serviços e as várias etapas envolvidas. Saber quais tarefas deverão ser executadas, levando em conta o nível de importância para a precedência, mão-de-obra especializada e equipamentos adequados disponíveis, é um problema combinatório complexo, classificado como NP-difícil (CRAVO, 2009) e muitos são os fatores que poderão influenciar no planejamento e execução.

Existem duas classes de métodos de resolução para os problemas de escalonamento: os métodos exatos e os métodos heurísticos. Como principal distinção entre os métodos, está que os heurísticos apresentam a característica de demandarem baixo tempo de execução, e em geral não há garantia de se alcançar a solução ótima. Já os métodos exatos, determinam a solução ótima, mas, comumente, demandam tempo de execução muito alto.

No entanto, esta premissa está mudando, uma vez que já é possível resolver problemas de grande porte com a utilização remota de *softwares* especializados, como por exemplo, através do *site neos-server.org* Server (NEOS GUIDE SERVER, 2017), processados em supercomputadores, como será explicado e aplicado neste trabalho.

Portanto, essa pesquisa tem como objetivo adaptar uma modelagem para o problema de escalonamento de tarefas, compreendendo tanto as alocações de equipes nas atividades do projeto quanto em definir a melhor ordem de execução dessas atividades, com o objetivo de gerar a melhor sequência de atividades, segundo as restrições impostas no problema. Minimizando a duração e o custo do projeto e alocando equipes mais qualificadas em cada atividade do projeto, através de um método de resolução exato.

Após a resolução do modelo adaptado, será realizada uma comparação com um modelo cuja função objetivo visa minimizar apenas o número de colaboradores, tendo com mesmas restrições as apresentadas no modelo anterior. O resultado será analisado conforme as necessidades do modelo em questão.

Será considerado um problema real de escalonamento de atividades pertencentes a um projeto de um Evento, realizado por uma Associação que promove eventos de cunho religioso e administra Igrejas Locais espalhadas pelo Brasil, contando com filiais fora do país. Os dados utilizados neste trabalho referem-se a um evento realizado durante os dias 27, 28, 29 e 30 de abril de 2018, com um público de mais de 2000 pessoas.

Este documento está organizado em cinco partes. Após a introdução, a segunda parte traz uma revisão da literatura nos assuntos que envolvem problemas de escalonamento de projetos, restrição de recursos e modelagem exata. Em seguida, na seção três, é detalhado estudo de caso e o problema a ser tratado, assim como o algoritmo utilizado para solucionar o problema proposto de forma exata, a adaptação que se fez necessária para sua aplicação ao modelo utilizado. Na quarta seção, são apresentados os dados usados para validar o procedimento, a solução computacional e a análise comparativa entre os modelos. Por fim, na seção cinco, são colocadas as conclusões do estudo desenvolvido.

1.1 OBJETIVOS

Para resolver a problemática de escalonamento de tarefas, propõe-se neste trabalho os seguintes objetivos.

1.1 Objetivo Geral

Aplicar e propor um modelo matemático exato para otimizar o escalonamento de tarefas entre colaboradores.

1.2 Objetivos Específicos

Para atingir o objetivo geral proposto neste trabalho, será necessário alcançar os seguintes objetivos específicos:

- Adquirir conhecimento sobre o tema escalonamento de tarefas de um projeto, esclarecendo definições, variedades e possíveis abordagens;
- Expor de forma detalhada, as atividades envolvidas no projeto abordado como problema a ser resolvido;
- Propor uma abordagem otimizada para o problema que envolve designação de tarefas e implementar um algoritmo que atenda às necessidades reais do problema;
- Comparar a resposta obtida com o cenário atual das equipes para as quais as tarefas serão designadas;
- Determinar equações matemáticas que descrevam as restrições impostas pelo projeto, através de um modelo matemático exato;
- Definir por meio de ferramentas computacionais soluções viáveis para o problema;
- Analisar os benefícios na organização através de um bom gerenciamento das atividades envolvidas no problema proposto.

1.2 JUSTIFICATIVA

A problemática a ser considerada neste trabalho envolve a execução de tarefas, compreendendo o escalonamento das mesmas aos colaboradores disponíveis para determinadas atividades do projeto de um evento de cunho religioso, promovido pela Associação Onda Dura. O surgimento do problema ocorreu durante a experiência do evento chamado “Conferência Atos 2018”, onde cerca de 87 colaboradores e mais de 40 voluntários, foram designados para a realização das tarefas descritas no projeto.

Os problemas de escalonamento de projetos com restrições de recursos têm ganhado a atenção dos pesquisadores, pois são modelos mais gerais que compreendem muitos dos problemas de escalonamento de máquinas, equipes, ou colaboradores, como casos especiais (BRUCKER et al., 1999; LEAL, 2007). Torna-se então, necessário a utilização de ferramentas estruturadas para suporte a estas questões e tomada de decisões.

Estas ferramentas necessitam de viabilidade em sua utilização. A aplicação de métodos e modelos matemáticos que reduzam custos, de maneira que não impactem na qualidade do produto final entregue ao cliente, é de grande interesse das organizações (BALLOU, 2006). Para tal, trazer para o cerne da pesquisa o conceito da otimização linear e mostrar como a utilização dessa ferramenta impacta diretamente na maneira como as empresas podem distribuir seus recursos e alocar suas tarefas, de forma a melhorar sua qualidade, otimizando o tempo e o custo das mesmas, podem ser passos decisivos para o aumento da competitividade e nível de excelência de uma organização.

Por outro lado, negar a possibilidade de uma otimização na disposição de recursos de um projeto, seria negligenciar o benefício resultante e a importância da aplicação da ferramenta de um modelo matemático para resolução de problemas específicos, com restrições próprias. O que significa uma aceleração no esgotamento de recursos ou agravamento de situações financeiras ou de tempo.

1.2.1 Acadêmica

A produção científica tem como objetivo apropriar-se da realidade para melhor analisá-la e, posteriormente, produzir transformações no cenário atuante. A discussão sobre os impactos do planejamento de tarefas (*scheduling*) nas etapas de quaisquer processos de produção ou execução de serviços, além de um aspecto prático muito relevante, reveste-se de importância para o meio acadêmico.

Nesse contexto, a produção de estudos e conteúdo que contribuam para a literatura disponível sobre modelagem de escalonamento de tarefas, com abordagem exata, impondo restrições adaptáveis para outros problemas conforme cenário real, justifica o desenvolvimento da pesquisa em questão.

1.2.2 Financeira

As condições de qualidade, produtividade e competitividade nos mercados interno e externo, assim como a busca pelo atendimento às necessidades dos clientes constituem os principais fatores responsáveis pela otimização do escalonamento de tarefas entre

colaboradores. No entanto, a falta de visão relacionada a aplicação dessa ferramenta, ainda retarda o aperfeiçoamento da mesma de forma prática.

Atualmente, determinar a melhor programação para escalonamento de tarefas tem sido um objetivo a ser alcançado por muitas organizações tendo em vista que, com isto, consegue-se a redução dos custos de um projeto, verificando o risco de multa por atraso na entrega, e a possibilidade de realocação de colaboradores, ou descarte dos mesmos, caso as tarefas possam ser realizadas com o mínimo de agentes.

Neste sentido a contribuição financeira desse trabalho se dá:

- À Associação Onda Dura, por encontrar uma sequência ótima de escalonamento de tarefas entre os colaboradores, reduzindo ou descartando a multa por atraso de execução e, portanto, aumentando seus lucros, ou reduzindo o custo destinado aos colaboradores;
- E aos participantes do evento, pois possibilita uma melhoria no serviço oferecido, e garantia do cumprimento de horários previstos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais conceitos relacionados ao tema deste trabalho, buscando gerar embasamento teórico quanto a pesquisa operacional, programação linear inteira e designação de tarefas, bem como o método utilizado para o desenvolvimento do estudo de caso, incluindo a abordagem de um modelo genérico com referência em um modelo de sequenciamento de tarefas em máquinas paralelas com tempo de processamento diferentes.

2.1 PROGRAMAÇÃO LINEAR INTEIRA

Segundo Goldbarg e Luna (2005) para a compreensão de todos os outros modelos existentes de programação matemática, é necessário compreender a programação linear como modelo pois seus conceitos serão extensíveis aos demais, servindo de base para estudos mais avançados. Além disso, é considerado extremamente eficiente nos algoritmos existentes, por ter alta capacidade de cálculo e podendo ser facilmente implementados por meio de planilhas, como por exemplo no *software* Excel.

Assim como descrito por Taha (2008):

Em PO, não temos uma única técnica para resolver todos os modelos matemáticos que podem surgir na prática. Em vez disso, o tipo e a complexidade do modelo matemático é que determinam a natureza do método de solução. A técnica mais utilizada de PO é a programação linear. Ela é aplicada a modelos cujas funções objetivo e restrição são lineares. Outras técnicas são a programação inteira (nas quais as variáveis assumem valores inteiros), a programação dinâmica (na qual o modelo original pode ser decomposto em subproblemas mais fáceis de tratar), a otimização em redes (na qual o problema pode ser modelado como uma rede) e a programação não linear (na qual as funções do modelo são não lineares).

Em geral, a Programação Linear trata dos problemas de alocações ótimas para recursos essencialmente escassos em determinados processos. Por mais abundantes que esses

recursos denominados escassos possam ser, representam na realidade a ideia de uma existência finita (COLIN, 2013).

Os problemas de Programação Linear, como qualquer outro da Pesquisa Operacional são compostos por três partes básicas: variáveis, objetivo e restrições. As variáveis, são comumente denominadas de variáveis de decisão e são os valores que procuramos determinar. Por outro lado, o objetivo, é o que efetivamente precisamos determinar, seja maximizando ou minimizando, e as restrições, são as limitações que o modelo está submetido e que devem ser satisfeitas pela solução final (TAHA, 2008). Sendo assim, o modelo matemático básico de um problema de PL pode ser representado pelas equações abaixo, formulado por Hillier e Lieberman (2013).

$$\text{Maximizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (2.1)$$

Sujeito às restrições:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \quad (2.2)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \quad (2.3)$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \quad (2.4)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \quad (2.5)$$

Em que a equação (2.1) representa o objetivo a ser otimizado (minimização ou maximização), as restrições (2.2) - (2.4) genéricas representam as restrições operacionais ou de capacidade a serem cumpridas, e as equações (2-5) definem o tipo de variáveis a serem consideradas. Ainda segundo Hillier e Lieberman (2013), para muitos problemas práticos as variáveis de decisões somente fazem sentido se possuírem valores inteiros. O modelo matemático de problemas de programação linear inteira (PLI) pode ser representado pelo modelo (2.1) -(2.5), acrescido da restrição de que as variáveis devem ter valores inteiros, ou seja, a restrição (2.6) também deverá ser considerada no modelo:

$$x_1, x_2, \dots, x_n \in Z. \quad (2.6)$$

Caso apenas algumas das variáveis tenham valores inteiros trata-se de um modelo denominado programação linear inteira mista (PLIM) (ARENALES, 2006). A programação linear inteira (PLI) é um ramo da PO que aborda métodos de otimização para problemas modelados através de um modelo linear inteiro, em que as variáveis pertencem ao conjunto dos números inteiros (ou discretos) (TAHA, 2008).

De acordo com Hillier e Lieberman(2013), problemas de PI tem um número bem menor de soluções a serem consideradas do que um problema de PL. Problemas de PI que contam com uma região de soluções limitadas, têm um número finito de soluções, porém esse número finito de soluções pode ser muito grande. Por exemplo, para um problema com n variáveis, há 2^n soluções consideráveis, portanto a cada uma variável que é adicionada a n , o número de soluções dobra, ou seja, há um crescimento exponencial na dificuldade do problema.

Esta classe de problemas possui fácil resolução quando se trata de problemas de pequeno porte. Quando a quantidade de variáveis aumenta, são necessários métodos heurísticos para a resolução, mas atualmente existem ferramentas computacionais desenvolvidas para resolver problemas complexos, fornecendo o resultado ótimo.

Nesta pesquisa, serão utilizados os conceitos de programação linear inteira (PLI) para modelar o problema de otimização do sequenciamento das tarefas entre os colaboradores em ambiente paralelo com tempo de execução das tarefas diferente por colaborador.

2.2 GESTÃO DE PROJETOS

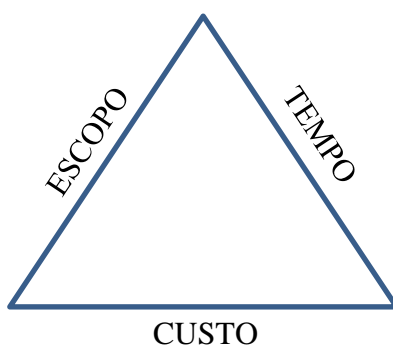
Desde tempos remotos, os homens realizam tarefas ou atividades que primeiramente idealizaram, conceberam e planejaram, sejam elas de implementação simples ou complexa. Um projeto pode ser definido como uma sequência de atividades (ou eventos) com início e fim estipulados, executadas segundo uma ordem previamente determinada, objetivando alcançar uma meta temporal pré-estabelecida. A estratégia para realiza-las de forma a potencializar o projeto e os recursos envolvidos, visando reduzir custos, é definida pela gestão do projeto.

De fato, o gerenciamento de projetos esteve sempre inerente à vivência humana, mas a consciencialização e reconhecimento da sua importância, teve o seu ápice na segunda metade do século XX, no período após a segunda guerra mundial. As fortes mudanças

macroeconômicas operadas nos modelos de desenvolvimento das nações ocidentais, caracterizados por forte incidência de concorrência, conduziram naturalmente as empresas a se abastecerem de meios, ferramentas e métodos que lhes permitissem ser mais produtivas e eficientes, ou seja, que lhes acrescentassem valor e dessa forma lhes garantissem sucesso. (LEAL, 2007).

Tavares (2002), define a Gestão de Projetos como sendo “o processo de conceber, preparar, organizar, dirigir e controlar as transformações necessárias para que se atinjam, a partir de um estado inicial S , os objetivos, definindo um novo estado S' , da forma mais conveniente”. Em uma abordagem considerada mais tradicional, tem por base o controle de três variáveis fundamentais, que podem ser entendidas como três vértices de um triângulo, estando intimamente relacionadas, conforme ilustrado na Figura 1.

Figura 1 - Base da Gestão de um Projeto



Fonte: Autoria própria, baseado em Tavares (2002)

Essas variáveis são o tempo de execução das atividades envolvidas em um projeto, o custo para desenvolvimento e execução das tarefas ou produtos e o escopo do projeto a ser gerenciado. O relacionamento existente entre as três variáveis é facilmente verificado, pois em regra, quanto maior for o objetivo do projeto, mais tempo ele levará a ser implementado e os custos tenderão a ser maiores. Numa outra perspectiva, se for implementar num período mais curto um determinado projeto, isso envolverá mais recursos e, portanto, resultará num custo maior. Por outro lado, será necessário garantir que existam esses mesmos recursos, ou que existam condições para que a sua aplicação se traduza realmente numa diminuição do tempo de realização.

Em suma, em cada projeto, existe sempre uma vertente financeira (custo), uma vertente organizacional (recursos humanos, materiais, espaço físico e temporal, entre outros) e uma vertente de controle que garanta que os objetivos do projeto serão atingidos.

Em uma outra abordagem, mais sistemática, o “*Project Management Institute - PMI*”, fundado em 1969 nos Estados Unidos, para normalizar as técnicas e ferramentas utilizadas na gestão de projetos enumera na sua publicação “*A Guide to the Project Management Body of Knowledge (PMBOK) - 5ª Edition*”, cinco fases de um projeto:

- Iniciação: processo de início de um projeto;
- Planejamento: atividade que envolve ferramentas e técnicas para realizar o projeto;
- Execução: processo que executa as etapas descritas no escopo do planejamento do projeto;
- Monitorização e controle: verificação do andamento e prazos do projeto;
- Encerramento: conclusão do projeto no término previamente definido.

Define ainda nove áreas de conhecimento que devem ser geridas no decorrer do projeto, associadas a estas fases:

- Integração do projeto;
- Escopo do projeto;
- Tempo do projeto;
- Custos do projeto;
- Qualidade do projeto;
- Recursos humanos do projeto;
- Comunicações do projeto;
- Riscos do projeto;
- Aquisições do projeto;
- Partes interessadas.

Claro que nem todos os projetos terão necessariamente que comportar todas as áreas acima citadas, cabendo ao gestor adaptar o modelo ao seu projeto em questão, de forma a

minimizar o risco associado e a maximizar as garantias de cumprimento dos objetivos definidos (LEAL, 2007).

Dentre as áreas citadas acima, no que se refere ao problema tratado neste trabalho, ele está relacionado de forma direta com o tempo de execução do projeto, custos, recursos humanos e qualidade do projeto como resultado final obtido. Por se tratar de um problema de escalonamento de tarefas, que visa otimizar a alocação das atividades entre os colaboradores envolvidos em um projeto, serão abordados alguns problemas de programação da produção para contextualizar acerca do tema a ser tratado.

O problema de escalonamento de tarefas e sua aplicabilidade é uma das áreas mais estudadas, por estar presente nos projetos desde o setor industrial ao setor dos serviços em geral. Exemplos frequentes deste tipo de problemas são: o planejamento de equipamentos produtivos na indústria, conhecido na terminologia anglo-saxônica como *production scheduling*, o planejamento de obras na construção civil ou a alocação de recursos a um projeto, citando apenas alguns (LEAL, 2007).

O tipo de problema está normalmente relacionado com o tipo de função objetivo que se pretende otimizar e com o tipo de restrições consideradas, como será apresentado a seguir. Um projeto bem definido terá sempre um, ou mais, objetivos específicos, definidos no modelo através de um determinado critério que se pretende otimizar.

2.2.1 Programação da Produção

A produção de qualquer produto ou serviço pode ser vista como um processo, ou um conjunto de processos, que tem como objetivo transformar insumos em produtos ou serviços. A execução, controle e planejamento das atividades de produção são de responsabilidade do Planejamento e Controle da Produção (PCP), o qual deve garantir que sejam produzidos os produtos certos, na quantidade certa e no tempo certo (SLACK et al., 2002).

O propósito do planejamento e controle da produção é garantir que o projeto ocorra de maneira eficaz e gere produtos e serviços da forma planejada, com a qualidade que os clientes esperam (SLACK, 2002). Para atingir tais objetivos de maneira eficaz, se faz necessário o desenvolvimento de uma programação da produção.

Graves (1981) definiu a programação da produção como sendo a alocação, no tempo, dos recursos disponíveis de produção de tal forma que satisfaça, da melhor maneira possível, um conjunto de critérios. Para ele, o problema de programação da produção envolve um conjunto de tarefas a serem realizadas e os critérios, que podem envolver decisões entre o término mais cedo e/ou mais tarde de cada uma delas.

A finalidade de um sistema de programação da produção é a utilização eficiente de recursos limitados na fabricação de produtos de modo a satisfazer a demanda dos clientes e gerar lucros para a empresa. As restrições incluem a disponibilidade de recursos e a satisfação dos clientes. A satisfação do cliente consiste em fornecer uma variedade de produtos a baixo custo e entregá-lo sem atraso e com qualidade (OLIVEIRA, 2002).

2.2.2 Problemas de Programação de Projetos

Programação de projetos é a aplicação de habilidades, técnicas e intuição adquiridas através do conhecimento e experiência para desenvolver modelos de programação eficaz. O modelo de programação integra e organiza logicamente componentes diversos do projeto, tais como atividades, recursos e relações lógicas, para aumentar a probabilidade da conclusão do projeto de sucesso dentro do período determinado (NEGREIRO, 2013).

O Problema de Programação de Projetos (PPP) consiste em encontrar um instante de início para todas as tarefas de forma a minimizar um ou mais objetivos, como, por exemplo, o instante do término do projeto (*makespan*) quando os recursos são limitados. Existe uma grande variedade de problemas de acordo com os elementos do projeto. Atividades podem ser executadas de diversos modos, como, por exemplo, um trabalhador em oito períodos de tempo ou quatro trabalhadores em dois períodos de tempo (ARENALES, 2007).

Brucker (1999), refere-se ainda à crescente importância deste tipo de problemas na atualidade, em virtude da generalizada tendência da adoção de metodologias *Lean* e *Make to order* pelas empresas. Os problemas de escalonamento de projetos são tratados no campo da Programação Inteira (PI) ou da Otimização Combinatória (OC).

Conforme Leal (2007), os recursos utilizados na execução das atividades podem ser classificados atendendo à categoria, ao seu tipo ou ao valor associado. Assim, os recursos dividem-se em três categorias:

- Recursos renováveis: limitados apenas num dado período. Após a sua utilização num dado período do horizonte do planeamento para a execução de uma determinada atividade, esses recursos voltam a estar disponíveis para nova utilização na mesma ou em outra atividade. Exemplos de recursos renováveis são as máquinas ou os recursos humanos;
- Recursos não renováveis: limitados em todo o horizonte do planeamento do projeto. Existe uma quantidade limitada de um determinado recurso que uma vez consumida não volta a estar disponível para outra utilização. O orçamento disponível para a execução de um projeto é um exemplo de recurso não renovável;
- Recursos duplamente restritos: recursos limitados num dado período e ao mesmo tempo no horizonte global de planeamento do projeto. Esses recursos podem sempre ser desdobrados em pelo menos um recurso renovável e num recurso não renovável. Um exemplo deste tipo de recursos poderá ser o dinheiro disponível para a realização global do projeto e que em cada período apenas está disponível um determinado montante.

No caso abordado neste trabalho, estará presente a utilização de recursos renováveis, se tratando dos recursos humanos, ou seja, dos colaboradores envolvidos na execução das atividades. E no que se trata da otimização, será considerado a utilização de um recurso duplamente restrito, que se enquadra como o recurso financeiro global do projeto, que deve ser minimizado.

Existem algumas nomenclaturas bastante utilizadas em trabalhos técnicos, que são definidas segundo Baker (1982), tais como:

- Tarefa – que também pode ser designada por ordem, produto, trabalho (*job*), corrida, atividade, entre outros. Pode ser um produto ou um lote de produtos idênticos, que devem ser processados pelos recursos produtivos;
- Máquina – que também pode ser posto, processador, equipamento entre outros, é o recurso produtivo destinado a executar uma operação. Outro recurso produtivo é o recurso humano que pode ser designado por homem, operadores ou colaboradores;
- Operação ou processamento – é o trabalho realizado pelo recurso produtivo sobre a tarefa;

- Tempo de processamento – é o tempo necessário para se concluir uma operação sobre uma tarefa. Neste trabalho, o tempo de processamento incluirá os tempos de preparação, transporte, colocação e retiradas dos recursos disponíveis para realizar as tarefas designadas;
- Roteiro – neste trabalho, roteiro será a sequência ordenada de recursos produtivos pelos quais devem passar as tarefas. Cada tarefa terá um roteiro independente;
- Programa – é uma sequência viável, onde estão especificados os horários de início e término de processamento de cada tarefa;
- Tempo ocioso – no presente trabalho, a referência a tempo ocioso estará associada ao intervalo de tempo em que um colaborador pode ficar parado quando, estando disponível e tendo uma tarefa para atender, ele aguarda para atender outra tarefa. Outros tempos em que os colaboradores possam estar parados, sem tarefa nenhuma na fila, não serão designados por tempo ocioso. Deste modo, inserção de tempo ocioso será fazer o colaborador ficar parado mesmo que tendo uma tarefa pronta para ser atendida na fila.

Dentre as nomenclaturas apresentadas anteriormente, serão utilizadas na descrição do modelo tratado no estudo de caso deste trabalho as seguintes, para melhor contextualização do problema, respectivamente:

- Tarefa;
- Colaboradores;
- Execução;
- Tempo de execução;
- Roteiro;
- Programa;
- Tempo Ocioso.

Para a classificação dos problemas de programação de projetos deve considerar fatores internos, que a influenciam por meio das requisições tecnológicas determinadas principalmente pelo fluxo padrão das tarefas nas máquinas, pelo número e tipos de máquinas disponíveis e pela precedência das tarefas (NAGANO, 1998).

O fluxo padrão consiste no direcionamento das tarefas com relação aos colaboradores, de tal forma que sempre utilize o mesmo colaborador para executar determinada tarefa num horário fixo. O número de colaboradores e a quantidade disponível para desempenhar as tarefas são importantes para a classificação da programação, pois materializam o tamanho da alocação dos recursos no tempo, permitindo a utilização específica dos métodos de otimização (SOUZA, 2009).

Para Nagano (1998) os principais problemas de programação encontrados na literatura são: *Job Shop*; *Open Job Shop*; *Batch Shop*; *Flow Shop*; *Batch/Flow Shop*; *Manufacturing Cells* (células de manufatura); *Assembly Shop* (estação de montagem); *Assembly Line* (linha de montagem); *Transfer Line* (linha de transferência); *Flexible Transfer Line* (linha de transferência Flexível). Este trabalho considera o problema do tipo *flow shop*.

2.2.3 Problema de Planejamento (*scheduling*)

Planejamento é um processo de otimização, no qual máquinas e recursos limitados são designados ao longo do tempo para atender diversas solicitações que são compostas de diversas atividades, denominadas tarefas. Esta designação é feita de tal maneira que sejam respeitadas integralmente as restrições de tempo de execução das atividades e o limite de capacidade do conjunto de máquinas e recursos usados por todas as atividades (PINEDO, 2005).

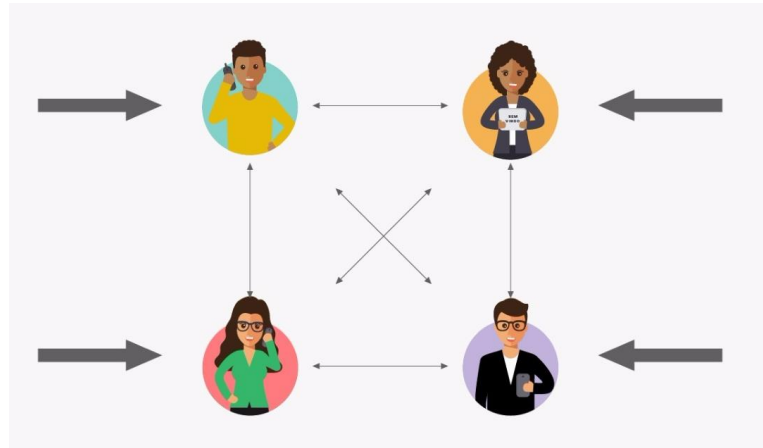
Baker (1997) afirma que *schedule* (ou escalonamento) caracteriza um plano tangível, tal como de horários de ônibus ou o de aulas. Usualmente um escalonamento informa quando certas coisas devem acontecer; ela mostra o plano de tempos de certas atividades e responde a questão: “quando alguma coisa terá lugar?”. A resposta para essa questão usualmente nos informa o horário (data), como por exemplo: uma aula de matemática se inicia às 13:00 h e finaliza às 17:00 h. Entretanto, outra resposta válida poderia ser em termos de sequência e não horário (data).

As principais decisões envolvidas nesse nível são: designação (sequenciamento) de tarefas (*jobs*), recursos e programação (planejamento) das tarefas em cada recurso, isto é, a sequência de processamento das tarefas e o instante de início e término do processamento de cada tarefa. Em geral, o problema de planejamento é caracterizado por três conjuntos: conjunto $T = \{T_1, T_2, \dots, T_n\}$ de n tarefas, conjunto $M = \{M_1, M_2, \dots, M_m\}$ de m máquinas e o conjunto $R = \{R_1, R_2, \dots, R_s\}$ de s tipos de recursos (BLAZEWICZ et al., 2007).

Os principais tipos de problema de planejamento são:

- *Job Shop* (Figura 2): n tarefas e m máquinas, em que cada tarefa é processada nas máquinas, de acordo com um roteiro e sem paradas das tarefas para execução de outras;

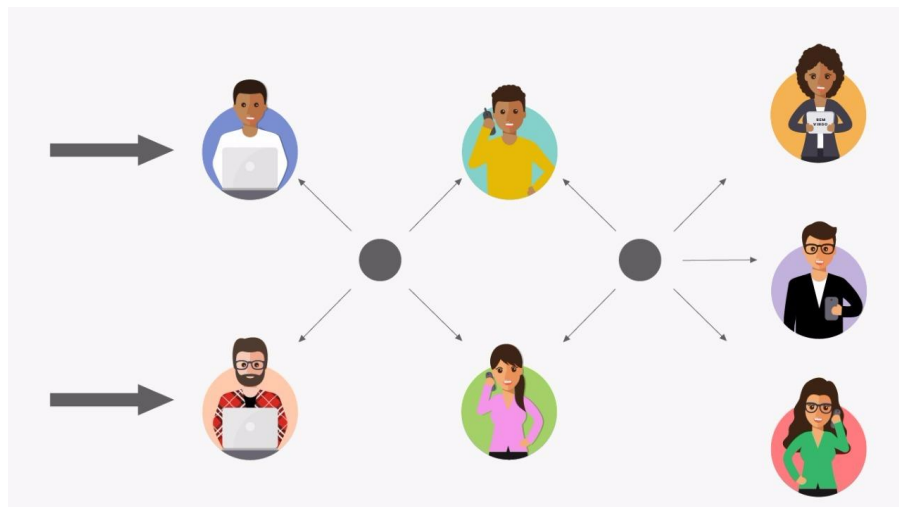
Figura 2 - Exemplo de processamento em máquinas em *Job Shop*



Fonte: Baseado em Santo (2013)

- *Open Shop* (Figura 3): Têm as mesmas características do *job shop*, com a permissão de parar tarefas para execução de outras e pode não haver relação de precedência entre as operações;

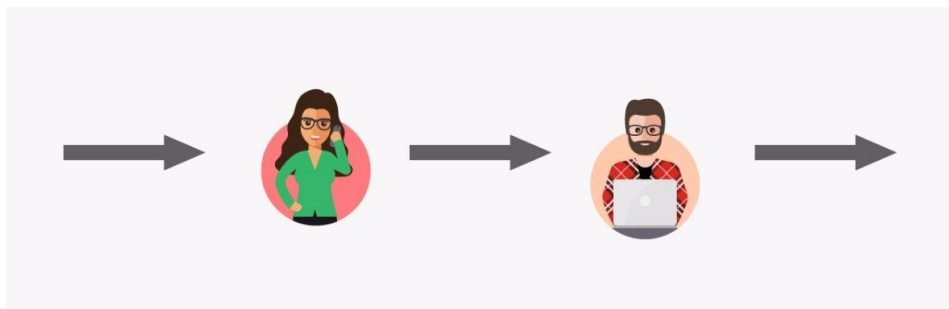
Figura 3 - Exemplo de processamento em máquinas em *Open Shop*



Fonte: Baseado em Santo (2013)

- *Flow Shop* (Figura 4): é um caso particular do *job shop*, em que as n tarefas têm o mesmo roteiro nas m máquinas (BLAZEWICZ et al., 2007; BRUCKER, 2007; ARENALES et al., 2007).

Figura 4 - Exemplo de processamento em máquinas em *Flow Shop*



Fonte: Baseado em Santo (2013)

Dois elementos são de fundamental importância para problemas de planejamento: as tarefas, que constituem cada uma das etapas do projeto a ser executado, e os recursos, que são os insumos necessários para que uma tarefa seja executada. As tarefas estão conectadas entre si através de relações de precedência que determinam a ordem em que as tarefas podem ou não ser executadas.

Normalmente, estas relações são do tipo *finish-to-start*, ou seja, é preciso que uma tarefa predecessora seja completamente executada antes de uma tarefa sucessora começar a ser. Também é muito comum que uma tarefa possa ter mais de uma predecessora. Neste caso, todas as predecessoras precisam ter sido executadas antes da tarefa em questão começar. O objetivo mais comum é fazer com que todas as tarefas do projeto sejam executadas o mais rapidamente possível, respeitando as restrições de precedência e de utilização dos recursos (SILVA, 2012).

2.2.4 Programação em *Flow Shop* (*Flow Shop Schedule*)

O problema de programação de *flow shop* (FSP) é um problema onde se mantém a mesma ordem de programação das n tarefas em todas as m máquinas. Basicamente com o objetivo de determinar entre as $n!$ possíveis sequências aquela que minimize alguma função objetivo estabelecida, tais como tempo total da programação das tarefas (*makespan*), tempo

ocioso das máquinas (*idletime*), tempo de fluxo das tarefas (*flowtime*), entre outros (GIGANTE, 2010).

Em um *flow shop*, segundo Baker (1997), cada tarefa tem sua própria sequência de processamento com fluxo linear unidirecional, ou seja, sem retorno no fluxo. O fluxo pode ser discreto, contínuo ou “semicontínuo”, cada tarefa só pode ser atendida por uma máquina de cada vez e cada máquina só pode atender uma tarefa por vez, não podendo a tarefa retornar para atendimento em máquinas anteriores.

De acordo com Gigante (2010) as hipóteses consideradas no problema de escalonamento de *flow shop* são:

- cada máquina está disponível continuamente, sem interrupções;
- cada máquina pode processar apenas uma tarefa de cada vez;
- cada tarefa pode ser processada por uma máquina de cada vez;
- os tempos de processamento das tarefas nas diversas máquinas são determinados e fixos;
- as tarefas têm a mesma data de liberação, a partir da qual, qualquer uma pode ser programada e executada;
- os tempos de preparação das operações nas diversas máquinas são incluídos nos tempos de processamento e independem da sequência de operações em cada máquina;
- uma vez iniciadas as operações nas diversas máquinas, elas não devem ser interrompidas.

2.3 O PROBLEMA DE ESCALONAMENTO DE TAREFAS

O escalonamento ou programação da produção é um dos problemas operacionais que visam planejar e controlar a execução das tarefas de produção e de serviços. Ele tem como finalidade determinar uma sequência factível de processamento de um conjunto de operações por um conjunto de recursos ao longo de um intervalo de tempo, visando otimizar uma ou mais medidas de desempenho.

Tais operações formam parte das tarefas ou pedidos de clientes por produtos ou serviços. Nesse problema podem existir ainda restrições de precedência entre as operações e

de disponibilidade de recursos por operação. As medidas de desempenho comumente usadas num problema de escalonamento pelo setor de planejamento operacional são:

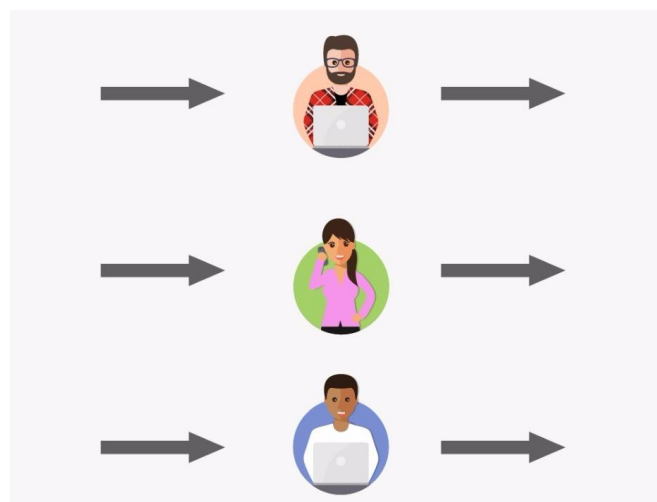
- *Makespan*: é a diferença de tempo entre o início e o fim de uma seqüência de trabalhos ou tarefas.
- Tempo de fluxo total: é o tempo total gasto por um conjunto de trabalhos ou tarefas, dado pelo somatório da conclusão de todos os trabalhos ou tarefas.
- Carga de trabalho total: representa o tempo total de trabalho utilizado pelos recursos ou máquinas.

Segundo Le Pape (2005) um problema de escalonamento consiste em decidir quando cada atividade deve ser executada, considerando as restrições de tempo e recurso. Para Hochbaum (1999) os objetivos em problemas de escalonamento são:

- Minimizar o tempo de execução total, medindo o nível de utilização das máquinas;
- Minimizar o tempo de espera de cada tarefa, que se trata do tempo entre a finalização de uma tarefa até o início da próxima;
- Minimizar os custos de execução das atividades.

O sequenciamento, corresponde à ordenação entre as diversas tarefas de um processo ou à ordem pela qual as tarefas serão processadas por determinado colaborador, conforme representado na Figura 5 (PINEDO, 2008).

Figura 5 - Escalonamento de tarefas entre colaboradores



Fonte: Baseado em Santo (2013)

Ainda em Pinedo (2008), nos problemas de escalonamento, tanto o número de tarefas quanto o número de máquinas são finitos. O número de tarefas é denota do por n e o número de máquinas por m . O autor estabelece ainda que, trata-se de um processo que visa definir a melhor sequência de processamento das tarefas, gerenciando a escassez de recursos ao longo do tempo. Portanto, é um processo de tomada de decisão que visa otimizar os objetivos estabelecidos.

A programação, para Pinedo (2008), atua na alocação de recursos para realização de tarefas em um determinado período de tempo, trata-se de um processo que envolve decisão com a meta de otimizar um ou mais objetivos. Esses objetivos podem ser diversos, como a redução do tempo total de processamento de um conjunto de tarefas, ou minimizar as tarefas completadas com atraso. É função do sequenciamento a busca pela adequação dos limitados recursos de produção, sejam eles, mão de obra, máquinas, instalações, materiais, entre outros. Isso dependerá do grau de detalhamento e intensidade que se dá a execução das 29 funções de sequenciamento no sistema produtivo em que se encontra inserido (TUBINO, 2007).

Segundo Arnold (1998), Scheduling é o processo definido por estabelecer as datas de começo e fim para as operações necessárias para a produção de um item. Para isso o planejador necessita de informações sobre o roteiro, capacidade produtiva, e tempo de processamento dos centros de trabalho envolvidos. Segundo Graves (1981), em Scheduling trabalha-se com operação no tempo de processamento, que é formado por cinco elementos:

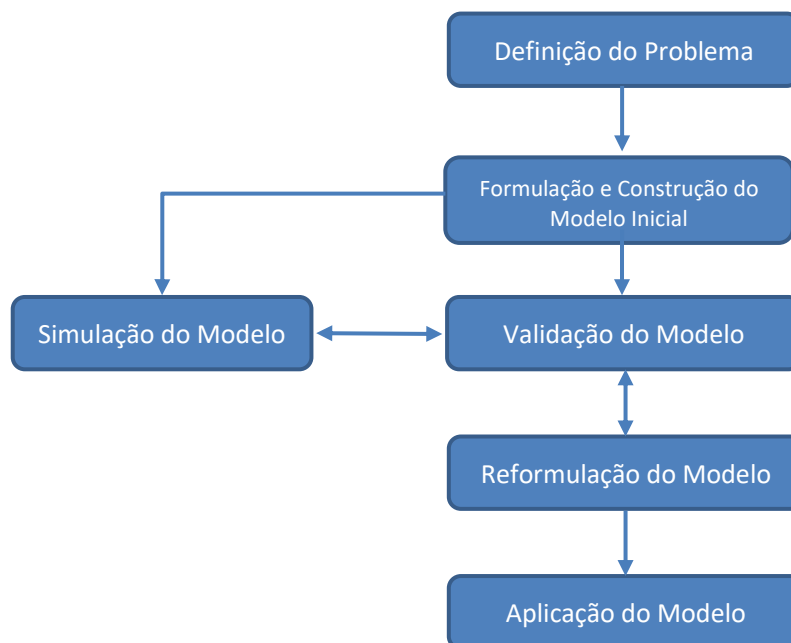
- Tempo em fila, é o tempo em que um trabalho aguarda para ser iniciado em um centro de produção;
- Tempo de setup, que é o tempo necessário para preparar um centro de trabalho para executar a operação;
- Tempo de corrida, é o tempo em que a operação é realizada;
- Tempo de espera, tempo de aguardo pelo material para ser transferido ao próximo centro de trabalho;
- Tempo de movimentação, tempo de trânsito entre os centros de trabalho.

Os objetivos do sequenciamento normalmente estão associados a regras de priorização, Tubino (2007) estabelece que essas regras pertencem a um procedimento simplificador que levam em consideração informações sobre tarefas e sobre as condições do sistema produtivo. Visa-se selecionar as tarefas que terão prioridade de processamento, alocando assim os recursos de maneira a executar essas tarefas.

Para aplicar o modelo encontrado ao problema definido como designação de tarefas em um Evento de grande escala, considera-se o problema de sequenciamento de n tarefas a serem realizadas em m máquinas paralelas, com tempo de processamento diferente, em lote com capacidade igual a K . Cada tarefa i possui um tamanho S_i , um tempo de processamento p_i . Então, um conjunto de tarefas poderá ser sequenciado em um lote se a soma de suas dimensões S_i for menor do que a capacidade K da máquina. Deve-se ater ao fato de que uma mesma tarefa não poderá ser sequenciada em mais de um lote e em mais de uma máquina. Cada tarefa i só será sequenciada em uma máquina, visando o aumento do processamento da quantidade das tarefas em um mesmo número de máquinas disponíveis. Contribuindo com a redução valor R_i no ganho final da empresa.

O objetivo do problema é definir o sequenciamento das tarefas nas máquinas. Nos modelos de otimização, não existem permissões de flexibilidade na escolha das alternativas, já que é estruturado para selecionar apenas uma das alternativas, que será chamada de ótima. Os critérios de escolha da solução ótima são pré-definidos e fazem parte da estrutura do modelo que através da análise matemática, busca a melhor alternativa. Os modelos de otimização são denominados mais especializados e tem a solução ótima como ponto de partida da decisão (ANDRADE, 2014).

Figura 6 -Processo de modelagem



Fonte: Autor Desconhecido.

Para melhor compreensão dos passos de construção do modelo, na Figura 6 exemplifica o processo de solução de um problema de pesquisa operacional. Conforme esquema a aplicação do modelo ocorrerá apenas após sua validação, realizada através dos dados gerados através do caso a ser otimizado.

Portanto, neste trabalho será considerado como base para formulação e construção de um modelo inicial de designação de tarefas, um modelo matemático exato de “Otimização do sequenciamento de tarefas em máquinas paralelas com tempos de processamento diferentes”, desenvolvido pela Engenheira de Transporte e Logística Larissa Bagini Barbosa, no ano de 2017, que será apresentado na seção a seguir.

2.4 SEQUENCIAMENTO DE TAREFAS EM MÁQUINAS PARALELAS

Para abordar o problema neste estudo, baseou-se em um modelo de sequenciamento de tarefas em várias máquinas paralelas proposto por Barboza (2017) e aplicado em um caso real de uma empresa multinacional do setor plástico.

A notação utilizada para a formulação do modelo é a seguinte:

- n - O número de tarefas a serem processadas;
- m - O número de máquinas a serem consideradas;
- p - O tempo de processamento da tarefa i na máquina k ;
- d - Prazo de execução das n tarefas. Ressalta-se que o prazo de execução é independente da máquina;
- $multa$ - Multa por atraso na execução da produção dos n produtos. A multa também é independente da máquina.

No qual definiu-se as seguintes variáveis:

- x_{ik} = Tempo de início, em horas, da tarefa i na máquina k a partir da hora zero,
 $\forall i = 1, \dots, n$ e $\forall k = 1, \dots, m$;
- $y_{ijk} = \begin{cases} 1, & \text{se a tarefa } i \text{ preceder a tarefa } j \text{ na máquina } k \\ 0, & \text{caso contrário} \end{cases}$,
 $\forall i, j = 1, \dots, n$, e $\forall k = 1, \dots, m$;
- $z_{ik} = \begin{cases} 1, & \text{se a tarefa } i \text{ é processada na máquina } k \\ 0, & \text{caso contrário} \end{cases}$,
 $\forall i = 1, \dots, n, \forall k = 1, \dots, m$.

E no modelo proposto em Barbosa (2017), considerou-se as seguintes restrições:

- Multa em caso de atraso no cumprimento do prazo, definido pela equação 2.7;
- Cada máquina pode processar apenas uma tarefa por vez, interruptamente, conforme definido nas equações 2.8 e 2.9;
- Cumprimento no prazo de execução, definido pelas equações 2.10 e 2.11;
- Todas as máquinas devem ser utilizadas, definido pela equação 2.12;
- Todas as tarefas devem ser processadas, definido pela equação 2.13;
- Cada tarefa pode ser processada por uma máquina e apenas uma vez, definido pela equação 2.14;
- Os tempos de processamento das tarefas nas diversas máquinas são determinados e fixos, definido pelas equações 2.15 e 2.16.

O modelo matemático para determinar o sequenciamento de tarefas que minimiza a multa por atraso no prazo de entrega das tarefas, completo, é então dado pelas equações (2.7-2.16).

$$\text{Minimizar } Z = \sum_{i=1}^n \sum_{k=1}^m R_i * s_{ik}^+ \quad (2.7)$$

S. a:

$$MM * y_{ijk} + (x_{ik} - x_{jk}) \geq p_{jk} \quad \forall i, j, k, i \neq j \quad (2.8)$$

$$MM * (1 - y_{ijk}) + (x_{jk} - x_{ik}) \geq p_{ik} \quad \forall i, j, k, i \neq j \quad (2.9)$$

$$x_{ik} + s_{ik}^- - s_{ik}^+ \leq d_i - p_{ik} + MM * yy_{ik} \quad \forall i, k \quad (2.10)$$

$$z_{ik} \leq MM * (1 - yy_{ik}) \quad \forall i, k \quad (2.11)$$

$$\sum_{i=1}^n z_{ik} \geq 1 \quad \forall k \quad (2.12)$$

$$\sum_{i=1}^n \sum_{k=1}^m z_{ik} = n \quad (2.13)$$

$$\sum_{k=1}^m z_{ik} = 1 \quad \forall i \quad (2.14)$$

$$s_{ik}^-, s_{ik}^+, x_{ik} \geq 0 \quad \forall i, k \quad (2.15)$$

$$y_{ijk}, y_{ik}, z_{ik} \in \{0,1\} \quad \forall i, j, k \quad (2.16)$$

Partindo do modelo de sequenciamento de tarefas para máquinas em paralelo proposto acima, o presente trabalho obteve seu desenvolvimento, conforme será explanado no próximo capítulo.

3 MÉTODO DE SOLUÇÃO

Inicialmente, neste capítulo, será apresentada a caracterização do problema a ser ponderado neste trabalho. Na sequência serão detalhados os procedimentos para a generalização do modelo para o escalonamento de tarefas entre vários colaboradores, e posteriormente, apresentado o método de solução.

Este capítulo apresenta e discute a modelagem matemática, tomando por base a modelagem apresentada por Barbosa (2017), que aborda a programação de tarefas em um sequenciamento de máquinas em paralelo.

Utilizaremos, portanto, as mesmas variáveis de decisão para definir o escalonamento das tarefas, como também os tempos de término da execução das tarefas por cada colaborador. No trabalho de Barbosa (2017) foram incluídas restrições para fazer com que todas as máquinas fossem utilizadas, sendo esta inutilizada neste problema. A função objetivo visa minimizar a quantidade de colaboradores que executarão as tarefas, sendo essa a diferença em relação ao trabalho de Barbosa (2017).

Abaixo segue as hipóteses utilizadas na programação:

- Cada colaborador estará disponível continuamente sem interrupções;
- Cada colaborador pode executar apenas uma tarefa de cada vez;
- Cada tarefa pode ser executada por um operador de cada vez;
- Os tempos de processamento das tarefas pelos colaboradores são determinados, fixos e diferem entre si;
- Determinadas tarefas podem ser realizadas exclusivamente por alguns colaboradores;
- As tarefas terão a mesma hora de liberação, a partir da qual, qualquer uma pode ser programada e executada;
- Para cada tarefa i haverá um intervalo de tempo, chamado de janela de tempo em que a tarefa não sofrerá nenhuma punição se seu horário de término estiver dentro deste intervalo, onde e_i é o tempo inicial para a tarefa e f_i é o tempo final para tarefa;
- Determinadas tarefas devem ser executadas antes de outras;

- Os tempos de preparação para a execução pelos diversos colaboradores são incluídos nos tempos de processamento e independem da sequência de execução por cada colaborador;
- As execuções dos diversos colaboradores, uma vez iniciadas, não devem ser interrompidas;
- Por ser um *flow shop*, o fluxo de processamento será unidirecional, deste modo, as tarefas não poderão retornar para máquinas anteriores no roteiro.

3.1 MODELO DE DESIGNAÇÃO DE TAREFAS PARA O ESTUDO DE CASO

O problema abordado no presente trabalho almeja otimizar o escalonamento das tarefas de um projeto, que envolve variáveis para cada tipo de tarefa entre cada colaborador. Particularmente, trata de atividades que forma o escopo de um evento, onde se leva em consideração os seguintes aspectos: o tempo de execução das tarefas, o número de colaboradores necessários para executá-las e a especificidade de cada colaborador.

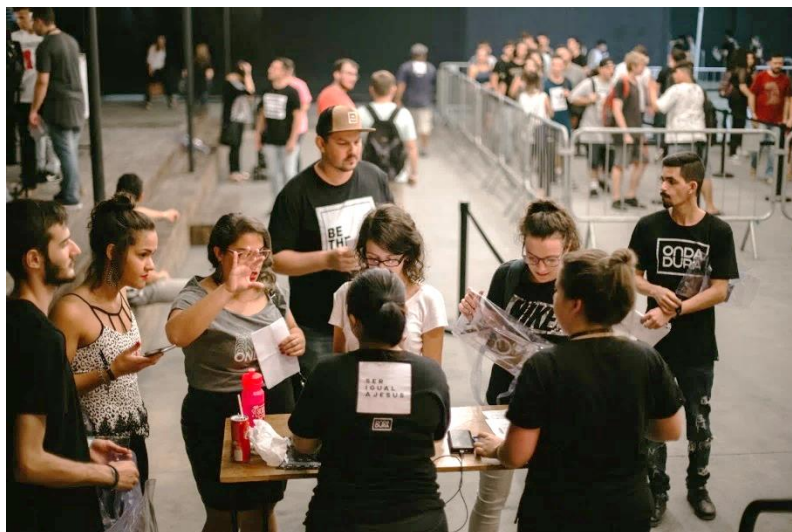
A motivação para a resolução deste problema se deu devido à ociosidade de equipes observados durante a montagem do evento “Conferência Atos 2018”, que ocorreu na Igreja Onda Dura em Joinville, onde o planejamento não era executado de modo otimizado, verificando-se por diversas vezes, colaboradores disponíveis no espaço de preparação do evento. Vale mencionar também que os colaboradores eram alocados para executar as tarefas em campo e, que muitas vezes aguardavam a conclusão de outras tarefas, para retornar as suas respectivas demandas.

Para o desenvolvimento do trabalho, foram consideradas informações verídicas de uma Associação que promove eventos de cunho religioso e administra Igrejas Locais espalhadas pelo Brasil, que conta com filiais fora do país.

Os dados utilizados neste trabalho referem-se a um evento realizado durante os dias 27, 28, 29 e 30 de abril de 2018, que comportou mais de 2000 pessoas.

As Figuras 7 e 8 são registros fotográficos dos dias do evento em questão.

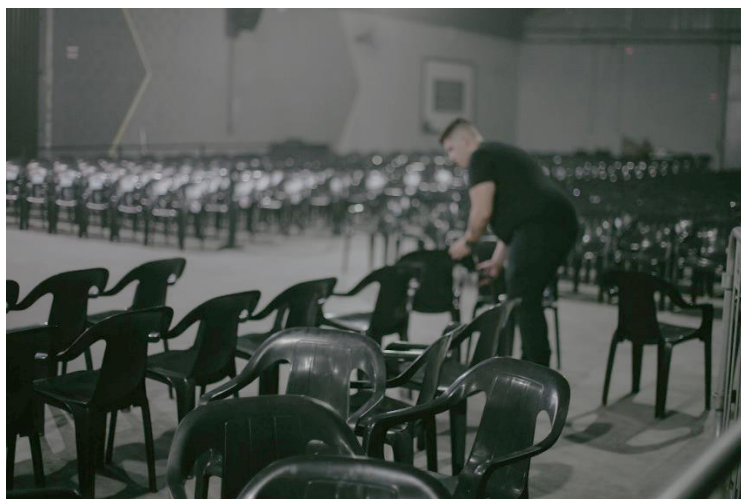
Figura 7 -Registro fotográfico do Evento “Conferência Atos 2018”



Fonte: Flickr Onda Dura, 2018.

O evento em questão contou com 87 colaboradores, que aqui serão representados no modelo como sendo operadores paralelos, que precisam realizar uma gama de 63 tarefas a serem realizadas em um período de 6 horas que antecedem o início do evento. Cada um dos colaboradores pode processar um conjunto de tarefas, e cada item possui um conjunto de colaboradores que podem realizar a tarefa. Cada tarefa corresponde a uma demanda a ser realizada, sendo que alguns itens de tarefa precisaram ser divididos em blocos para serem considerados como um item de tarefa no modelo.

Figura 8 - Registro fotográfico do Evento “Conferência Atos 2018”

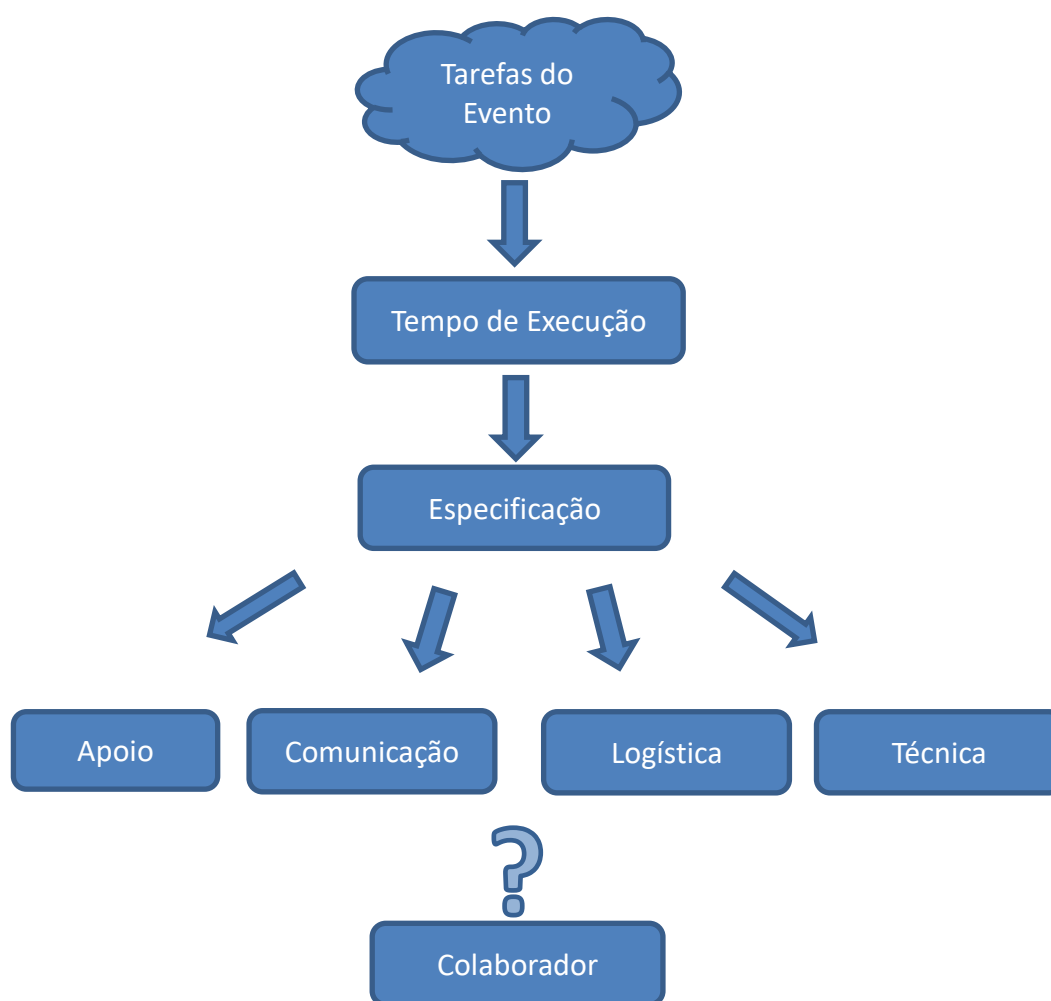


Fonte: Flickr Onda Dura, 2018.

Ressalta-se que algumas tarefas podem somente ser realizadas por alguns colaboradores e o tempo de execução de uma mesma tarefa pode variar dependendo de qual colaborador for realizá-la. Algumas tarefas precisam ser executadas antes de outras, conforme o planejamento operacional do evento em questão.

A Figura 9 demonstra graficamente como é inicializada a solicitação da execução das tarefas. Inicialmente, as tarefas são listadas conforme necessidades do planejamento do projeto, através de uma ordem de precedência de execução entre elas, com diferentes tempos de execução para cada colaborador, onde determinadas tarefas podem ser realizadas por determinados colaboradores.

Figura 9 - Processo de Decisão para Escalonamento de Tarefa



Fonte: Autoria Própria (2018)

Com o objetivo de reduzir a multa por atraso conforme o modelo inicial, encontrar a melhor forma para sequenciar as tarefas e qual colaborador deve executá-la. Para que ao final da solução do problema se possa ter um parâmetro de tomada de decisão para reduzir ou não o número de colaboradores que serão escalados para o mesmo evento no próximo ano.

Inicialmente procurou-se adaptar, testar e validar, com o intuito de apresentar um novo modelo que atenda o problema proposto, partindo do modelo apresentado no trabalho “Otimização do sequenciamento de tarefas em máquinas paralelas com tempos de processamento diferentes” (BARBOSA, 2017), apresentado de forma mais detalhada na Fundamentação Teórica.

Os seguintes parâmetros foram considerados:

- n - O número de tarefas a serem processadas;
- m - O número de colaboradores a serem considerados;
- t - O tempo de processamento da tarefa i na máquina k ;
- p - Prazo de execução das n tarefas. Ressalta-se que o prazo de execução independe do colaborador;
- *Multa* - Multa por atraso na execução das n tarefas. A multa também independe do colaborador que realizar a tarefa.

A notação utilizada para a formulação foi a seguinte:

- MM – é um número suficientemente grande (para restringir a escolha dos colaboradores para determinadas tarefas);
- i/j – índice para as tarefas;
- k – índice para colaborador.

3.1.1 Definição das variáveis

Definiu-se as seguintes variáveis:

- x_{ik} = Tempo de início, em minutos, da tarefa i no colaborador k a partir do minuto zero, $\forall i = 1, \dots, n$ e $\forall k = 1, \dots, m$;
- $y_{ij} = \begin{cases} 1 & \text{se a tarefa } i \text{ executada pelo colaborador } j \\ 0, & \text{caso contrário} \end{cases}$,

$$\forall i = 1, \dots, n \text{ e } \forall k = 1, \dots, m;$$

$$\bullet \ y_{ijk} = \begin{cases} 1 & \text{se a tarefa } i \text{ preceder a tarefa } j \text{ no colaborador } k \\ 0 & \text{caso contrário} \end{cases},$$

$$\forall i = 1, \dots, n \text{ e } \forall k = 1, \dots, m;$$

$$\bullet \ z_{ik} = \begin{cases} 1 & \text{se a tarefa } i \text{ é executada pelo colaborador } k \\ 0 & \text{caso contrário} \end{cases},$$

$$\forall i = 1, \dots, n \text{ e } \forall k = 1, \dots, m;$$

3.1.2 Definição das Restrições

Nessa seção será abordado de forma detalhada cada restrição imposta no modelo matemático de otimização para escalonamento de tarefas. Sendo que entre elas estão restrições definidas e propostas por Barbosa (2017), e também as restrições designadas ao problema proposto.

3.1.2.1 Restrições de não interferência entre as tarefas

As restrições de não interferência garantem que duas tarefas distintas i e j , com tempos de processamento pelo colaborador k dados por p_{ik} e p_{jk} , respectivamente, não serão executadas simultaneamente pelo colaborador k , se o tempo de início de processamento da tarefa i pelo colaborador k é maior ou igual ao tempo de início de processamento da tarefa j acrescido do tempo de processamento da tarefa j pelo colaborador k , que matematicamente pode ser expresso por:

$$x_{ik} \geq x_{jk} + p_{jk}, \text{ se a tarefa } i \text{ preceder } j \text{ no colaborador } k, \quad (3.1)$$

ou, se o tempo de início de processamento da tarefa j no colaborador k é maior ou igual ao tempo de início de processamento da tarefa i no colaborador k , acrescido do tempo de processamento da tarefa i no colaborador k , ou seja, se:

$$x_{jk} \geq x_{ik} + p_{ik}. \quad (3.2)$$

Para transformar essas restrições (ou-ou) em restrições simultâneas definiu-se a variável binária auxiliar:

$$y_{ijk} = \begin{cases} 1, & \text{se } i \text{ preceder } j \\ 0, & \text{se } j \text{ preceder } i \end{cases}, \text{ no colaborador } k. \quad (3.3)$$

E considerou-se uma constante MM positiva grande, fazendo-se:

$$MM y_{ijk} + (x_{ik} - x_{jk}) \geq p_{jk} \quad (3.4)$$

e

$$MM(1 - y_{ijk}) + (x_{jk} - x_{ik}) \geq p_{ik}. \quad (3.5)$$

Essa conversão garante que apenas uma das duas restrições (3.4) e (3.5) pode ser ativada.

3.1.2.2 Restrição de Cobertura

Para garantir que todas as tarefas sejam executadas (cobertas), considera-se:

$$\sum_{i=1}^n \sum_{k=1}^m z_{ik} = n \quad (3.6)$$

3.1.2.3 Restrição de Empacotamento

Para garantir que cada tarefa seja processada apenas uma vez, considera-se:

$$\sum_{k=1}^m z_{ik} = 1, \forall i = 1, \dots, n. \quad (3.7)$$

3.1.2.4 Restrições de Cumprimento do Prazo de Execução

Para garantir a restrição de cumprimento de prazo de execução, considerando-se s_{jk} uma variável irrestrita, o prazo de execução da tarefa j pode ser dado por:

$$d_j = x_{jk} + p_{jk} + s_{jk}. \quad (3.8)$$

Quando $s_{jk} \geq 0$, o prazo é cumprido, e se $s_{jk} < 0$, a multa é aplicada pelo atraso.

Considerando a substituição:

$$s_{jk} = s_{jk}^- - s_{jk}^+, \quad (3.9)$$

com $s_{jk}^-, s_{jk}^+ \geq 0$, então a restrição (3.8) se torna:

$$x_{jk} + s_{jk}^- - s_{jk}^+ \leq d_j - p_{kj}. \quad (3.10)$$

Com isso, o custo da multa por atraso é proporcional a s_{jk}^+ .

No entanto, a restrição (3.9) só deverá valer na máquina k quando a tarefa j for executada nesta máquina. Matematicamente, essa condição é equivalente a dizer que se,

$$z_{ik} > 0, \quad (3.11)$$

então,

$$x_{jk} + s_{jk}^- - s_{jk}^+ \leq d_j - p_{kj}. \quad (3.12)$$

As restrições (se-então) devem ser transformadas em restrições simultâneas. Para isso, considerando-se a constante $MM > 0$, e definindo-se a variável binária auxiliar:

$$yy_{ik} = \begin{cases} 1, & z_{ik} > 0 \\ 0, & \text{caso contrário, } \forall i = 1, \dots, n \text{ e } \forall k = 1, \dots, m. \end{cases} \quad (3.13)$$

Pode-se escrever as seguintes restrições simultâneas:

$$x_{jk} + s_{jk}^- - s_{jk}^+ \leq d_j - p_{kj} + Myy_{ik}, \quad (3.14)$$

$$z_{ik} \leq M * (1 - yy_{ik}). \quad (3.15)$$

As restrições (3.14) e (3.15) garantem que se a tarefa j for executada no colaborador k , então o tempo de execução pelo colaborador deverá ser satisfeito.

3.1.2.5 Restrição de Precedência entre as tarefas

Para garantir que determinadas tarefas, especificadas, sejam executadas antes de outras, conforme necessidade do problema, e ordem de tarefas fornecidas pelos dados, se fez necessário incrementar a seguinte restrição:

$$x_{ik} - x_{i-1,k} \geq t_{ik} \quad (3.17)$$

Sendo que a equação 3.17, no modelo pode ser repetida conforme necessidades de mais de uma tarefa ser precedente de outra no modelo, conforme dados fornecidos.

3.1.2.6 Multa por atraso

O custo da multa por atraso pode ser dado por:

$$\sum_{i=1}^n \sum_{k=1}^m R_i * s_{ik}^+ \quad (3.16)$$

3.1.3 Modelo Algébrico considerando n tarefas e m colaboradores

O modelo matemático para determinar o escalonamento de tarefas que minimiza a multa por atraso no prazo de entrega das tarefas é então dado pelas equações (3.20-3.29).

$$\text{Minimizar } Z = \sum_{i=1}^n \sum_{k=1}^m R_i * s_{ik}^+ \quad (3.20)$$

S. a:

$$MM * y_{ijk} + (x_{ik} - x_{jk}) \geq p_{jk} \quad \forall i, j, k, i \neq j \quad (3.21)$$

$$MM * (1 - y_{ijk}) + (x_{jk} - x_{ik}) \geq p_{ik} \quad \forall i, j, k, i \neq j \quad (3.22)$$

$$x_{ik} + s_{ik}^- - s_{ik}^+ \leq d_i - p_{ik} + MM * yy_{ik} \quad \forall i, k \quad (3.23)$$

$$z_{ik} \leq MM * (1 - yy_{ik}) \quad \forall i, k \quad (3.24)$$

$$\sum_{i=1}^n \sum_{k=1}^m z_{ik} = n \quad (3.25)$$

$$\sum_{k=1}^m z_{ik} = 1 \quad \forall i \quad (3.26)$$

$$x_{ik} - x_{i-1,k} \leq t_{ik} \quad (3.27)$$

$$s_{ik}^-, s_{ik}^+, x_{ik} \geq 0 \quad \forall i, k \quad (3.28)$$

$$y_{ijk}, yy_{ik}, z_{ik} \in \{0,1\} \quad \forall i, j, k \quad (3.29)$$

3.2 MODELO PARA OTIMIZAÇÃO DO NÚMERO DE COLABORADORES

Considerando a hipótese de a organização não necessariamente precisar minimizar a multa, que foi inserida com o intuito de designar pesos de prioridades para cada tarefa, adaptou-se o modelo para que sua resolução obtivesse o número mínimo de colaboradores para que todas as tarefas sejam executadas.

Nessa seção serão apresentadas as variáveis e as restrições inseridas no modelo anteriormente apresentado, para que o novo objetivo proposto fosse alcançado.

3.2.1 Definição de Variáveis

Além das variáveis definidas no modelo da seção anterior, definiu-se:

$$\bullet \quad op_k = \begin{cases} 1 & \text{se o colaborador } k \text{ executar alguma tarefa} \\ 0 & \text{caso contrário} \end{cases},$$

$$\forall k = 1, \dots, m.$$

3.2.2 Definição das Restrições

Nessa seção serão abordadas de forma detalhada as restrições impostas no modelo matemático de otimização do número de colaboradores.

3.2.2.1 Restrição de garantia do Colaborador executar alguma tarefa

Para garantir que determinados colaboradores sejam determinados para executar mais de uma tarefa, tornando possível a minimização do número total de colaboradores:

$$op_k \geq z_{ik}, \forall i = 1, \dots, n. \quad (3.30)$$

3.2.2.2 Restrição para limitar o número de Colaboradores

A restrição de um contador garante que um mesmo colaborador possa executar mais de uma tarefa, e ainda assim ser contabilizado apenas uma vez na busca da quantidade mínima total de colaboradores:

$$op_k \leq \sum_{i=1}^n z_{ik} \quad (3.31)$$

3.2.2.3 Restrição de capacidade de tarefas por colaborador

Para garantir que todas as tarefas sejam realizadas em 360 minutos, que é o período de montagem do evento:

$$\sum_{i=1}^n t_{ik} \times z_{ik} \leq 360 \quad (3.32)$$

3.2.2.4 Quantidade de Colaboradores

A busca pela quantidade mínima de colaboradores para executar as tarefas considera a seguinte função objetivo:

$$\text{Minimizar } Z = \sum_{k=1}^m op_k \quad (3.33)$$

3.2.3 Modelo Algébrico para otimização dos colaboradores

O modelo matemático completo para determinar o escalonamento de tarefas, que minimiza a quantidade de colaboradores necessários para a execução de todas as tarefas, é então dado pelas equações (3.33-3.45).

$$\text{Minimizar } Z = \sum_{k=1}^m op_k \quad (3.34)$$

S. a:

$$MM * y_{ijk} + (x_{ik} - x_{jk}) \geq t_{jk} \quad \forall i, j, k, i \neq j \quad (3.35)$$

$$MM * (1 - y_{ijk}) + (x_{jk} - x_{ik}) \geq t_{ik} \quad \forall i, j, k, i \neq j \quad (3.36)$$

$$x_{ik} + s_{ik}^- - s_{ik}^+ = p_i - t_{ik} + MM * yy_{ik} \quad \forall i, k \quad (3.37)$$

$$z_{ik} \leq MM * (1 - yy_{ik}) \quad \forall i, k \quad (3.38)$$

$$\sum_{i=1}^n \sum_{k=1}^m z_{ik} = n \quad (3.39)$$

$$\sum_{k=1}^m z_{ik} = 1 \quad \forall i \quad (3.40)$$

$$op_k \geq z_{ik} \quad \forall i, k \quad (3.41)$$

$$op_k \leq \sum_{i=1}^n z_{ik} \quad \forall k \quad (3.42)$$

$$\sum_{i=1}^n t_{ik} \times z_{ik} \leq 360 \quad \forall k \quad (3.43)$$

$$s_{ik}^-, s_{ik}^+, x_{ik} \geq 0 \quad \forall i, k \quad (3.44)$$

$$y_{ijk}, yy_{ik}, z_{ik} \in \{0,1\} \quad \forall i, j, k \quad (3.45)$$

No próximo capítulo será apresentada a aplicação dos modelos propostos considerando-se os dados obtidos.

4 RESULTADOS COMPUTACIONAIS

Neste capítulo, serão abordados os resultados obtidos com a implementação dos métodos propostos, através do experimento computacional realizado para solucionar o problema de escalonamento de tarefas.

Inicialmente, será detalhado o cenário do estudo de caso e em seguida os resultados com o uso da otimização do escalonamento com modelo exato. Além disso será realizada uma análise comparativa, entre a viabilidade de manter a quantidade de colaboradores envolvidos no projeto, obtida com o uso dos métodos e a utilizada atualmente na organização.

4.1 DETERMINAÇÃO DAS AMOSTRAS DO ESTUDO DE CASO

Para o experimento computacional foram utilizados a média dos tempos de execução das tarefas de 87 colaboradores observados durante o evento realizado. Os colaboradores foram divididos em quatro equipes, para melhor organização e visualização dos dados, sendo estes:

- Apoio: formada por 40 colaboradores, tal conjunto não executa nenhuma tarefa específica;
- Técnica: formada por 12 colaboradores, dos quais 7 fazem parte da banda e são os únicos que podem realizar a tarefa “passagem de som”. E 5 técnicos que são designados para realizar as tarefas que exigem competências técnicas;
- Comunicação: formada por 10 colaboradores, dos quais apenas estes podem executar tarefas que envolvem questões de identidade visual e design;
- Logística: formada por 25 colaboradores, que preferencialmente realizam as tarefas de montagem dos ambientes externos.

O conjunto de dados de cada colaborador encontra-se no Apêndice A, de maneira sintetizada, conforme utilizado no modelo final. Sua versão completa pode ser acessada de forma online, disponível em: <https://drive.google.com/file/d/1wvpdAqwfvuBAZwowrvOvYHSzLppGR4Gi/view?usp=sharing>.

Não foi possível utilizar todos os dados, devido o *solver* não comportar tamanha entrada. Por conta disso os dados de entrada foram por amostra de cada equipe apresentada, com uma porcentagem proporcional ao tamanho das equipes a cada teste.

4.2 IMPLEMENTAÇÃO

Utilizou-se a ferramenta *Solver* Gurobi em AMPL, versão 5.0, do site *NEOS Server: State-of-Art Solver for Numerical optimization* (www.neos-server.org) (NEOS SERVER GUIDE, 2014). Trata-se de um *site*, de acesso livre, para solução de problemas de otimização numérica sem limite de variáveis. Com suporte da *University of Wisconsin*, em Madison, os modelos matemáticos são executados remotamente em máquinas de alto desempenho distribuídos em várias universidades pelo mundo como por exemplo na *Arizona State University*, *University of Klagenfurt* na Austria, e *University of Minho* em Portugal (NEOS SERVER GUIDE, 2014).

As etapas, explicadas passo a passo, de como executar um modelo matemático remotamente no *NEOS Server*, estão disponíveis em Barbosa (2017).

A seguir serão apresentados os dois modelos e suas resoluções em AMPL.

4.2.1 Modelo de busca pela redução da Multa

O modelo em AMPL a ser resolvido utilizando o *Solver* GUROBI, pode ser observado a seguir na Figura 10.

Figura 10 - Modelo completo em AMPL

```
----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n, k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n, k in 1..m} ≥ 0;
var y{i in 1..n, j in 1..n, k in 1..m} binary;
var yy{i in 1..n, j in 1..m} binary;
```

```

var z{i in 1..n, k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{i in 1..n} R[i] * sP[i];
subject to C1 {i in 1..n, j in 1..n, k in 1..m : i ≠ j}: MM * y[i, j, k] + (x[i, k] - x[j, k]) ≥ t[j, k];
subject to C2 {i in 1..n, j in 1..n, k in 1..m : i ≠ j}: MM * (1 - y[i, j, k]) + (x[j, k] - x[i, k]) ≥ t[i, k];
subject to C3 {i in 1..n, k in 1..m}: x[i, k] + sM[i, k] - sP[i, k] = p[i] - t[i, k] + MM * yy[i, k];
subject to C4 {i in 1..n, k in 1..m}: z[i, k] ≤ MM * (1 - yy[i, k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z[i, k] = 1;
subject to C6 : sum{i in 1..n, k in 1..m} z[i, k] = n;
subject to C7 {k in 1..m}: x[i - 1, k] - x[i, k] ≤ -t[1 - 1, k];

```

Fonte: Baseado em Barbosa, 2017

4.2.2 Modelo de busca pela redução de Colaboradores

O modelo em AMPL a ser resolvido utilizando o *Solver* GUROBI, pode ser observado a seguir na Figura 11.

Figura 11 - Modelo completo em AMPL

```

----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n, k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n, k in 1..m} ≥ 0;
var y{i in 1..n, j in 1..n, k in 1..m} binary;
var yy{i in 1..n, j in 1..m} binary;
var z{i in 1..n, k in 1..m} binary;
var op{i in 1..n, k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{k in 1..m} op[k];
subject to C1 {i in 1..n, j in 1..n, k in 1..m : i ≠ j}: MM * y[i, j, k] + (x[i, k] - x[j, k]) ≥ t[j, k];
subject to C2 {i in 1..n, j in 1..n, k in 1..m : i ≠ j}: MM * (1 - y[i, j, k]) + (x[j, k] - x[i, k]) ≥ t[i, k];
subject to C3 {i in 1..n, k in 1..m}: x[i, k] + sM[i, k] - sP[i, k] = p[i] - t[i, k] + MM * yy[i, k];
subject to C4 {i in 1..n, k in 1..m}: z[i, k] ≤ MM * (1 - yy[i, k]);

```

```

subject to C5 {i in 1..n}: sum {k in 1..m} z[i,k] = 1;
subject to C6 : sum{i in 1..n, k in 1..m} z[i,k] = n;
subject to C7 {i in 1..n, k in 1..m}: op[k] ≥ z[i,k];
subject to C8 {k in 1..m} : op[k] ≤ sum{i in 1..n} z[i,k];
subject to C9 {k in 1..m}: sum{i in 1..n} z[i,k] * t[i,k] ≤ 360;

```

Fonte: Baseado em Barbosa, 2017

4.2.3 Dados para entrada dos modelos

Logo após é necessário o arquivo com os dados a serem utilizados no modelo em AMPL. Devido serem muitos dados, consta na Figura 12, a seguir o arquivo de dados de entrada da Tabela sintetizada.

Figura 12 - Modelo sintetizado em AMPL arquivo de dados

```

----- modelo para m colaboradores-----
param n = 63;
param m = 87;
param MM = 100000;
param t:
1  2  3  4  5  6  7  8  9  10  11  12  13... m :=
1  45  46  47  48  49  50  51  52  53  54  45  48... m
2  46  47  48  49  50  51  52  53  54  55  49  54... m
3  47  48  49  50  51  52  53  54  55  56  50  45... m
4  48  49  50  51  52  53  54  55  56  57  44  51... m
5  49  50  51  52  53  54  55  56  57  58  45  48... m
6  50  51  52  53  54  55  56  57  58  59  46  47... m
7  51  52  53  54  55  56  57  58  59  60  48  53... m
8  52  53  54  55  56  57  58  59  60  61  56  59... m
9  32  25  39  20  21  22  23  24  25  26  30  38... m
:  :  :  :  :  :  :  :  :  :  :  :  :
n  ... n;

param p: 1  2  3  4  5  6  7  8  9  10  11  12  13 ... n :=
1  180  180  180  180  180  180  180  180  180  120  120  120  120  120 ... n;

param R: 1  2  3  4  5  6  7  8  9  10  11  12  13 ... n :=
1  50  50  50  50  50  50  50  50  50  40  40  40  40  40 ... n;

```

Fonte: Baseado em Barbosa, 2017

4.3 SOLUÇÃO PROPOSTA PARA REDUÇÃO DA MULTA

Inicialmente o modelo foi implementado com os dados completos do problema, porém o *Software* Gurobi, versão 7.5.1, retornou “erro”, devido exceder a memória máxima atribuída ao arquivo submetido.

4.3.1 Modelo Algébrico considerando 4 tarefas e 3 colaboradores

Exclusivamente para fins de validação do modelo com precedência de tarefas, foi proposta essa solução para então se expandir o modelo e suas restrições de precedência. Quando não for cumprido o prazo de execução da tarefa incide-se uma multa proporcionalmente ao tempo de atraso.

O valor da multa não se relaciona com o problema de forma financeira, mas sim atribuindo a cada tarefa um grau de prioridade a ser executada o quanto antes for possível, respeitando as demais restrições.

A Tabela 1 apresenta os tempos de execução, intervalo na janela de tempo em que as tarefas podem ser executadas, denominados aqui como prazos e os valores da multa por atraso.

Tabela 1 - Dados para escalonamento de 4 tarefas entre 3 colaboradores

Tarefas	Prazo de execução (minutos)	Tempo de Execução (minutos)			Multa (grau)
		Colaborador 1	Colaborador 2	Colaborador 3	
1	60	49	45	56	50
2	120	28	33	40	40
3	200	-	-	180	60
4	360	190	185	-	30

Fonte: Autoria própria

Ressalta-se que para os dados apresentados na Tabela 1, a Tarefa 1 deve ser realizada antes da Tarefa 2, obrigatoriamente, se tornando então uma restrição de precedência de tarefa. O arquivo para a entrada de dados está representado na Figura 14. Utiliza-se um valor grande (999), muito maior que todos os outros valores utilizados como tempo de execução, para os itens que não podem ser executados pelo colaborador k , a fim de proibir seu escalonamento em um colaborador que não poderá realizar a tarefa.

Outra especificidade para esse modelo é que a Tarefa 3 pode ser realizada apenas pelo Colaborador 3, e a Tarefa 4 não pode ser realizada por esse mesmo Colaborador. O modelo em AMPL a ser resolvido utilizando o Solver GUROBI, pode ser observado na Figura 13 e 14.

Figura 13 - Modelo para 4 tarefas e 3 colaboradores em AMPL

```

----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n, k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n, k in 1..m} ≥ 0;
var y{i in 1..n, j in 1..n, k in 1..m} binary;
var yy{i in 1..n, j in 1..m} binary;
var z{i in 1..n, k in 1..m} binary;
var o{i in 1..n, k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{i in 1..n} R[i] * sP[i];
subject to C1 {i in 1..n, j in 1..n, k in 1..m : i != j}: MM * y[i, j, k] + (x[i, k] - x[j, k]) ≥ t[j, k];
subject to C2 {i in 1..n, j in 1..n, k in 1..m : i != j}: MM * (1 - y[i, j, k]) + (x[j, k] - x[i, k]) ≥ t[i, k];
subject to C3 {i in 1..n, k in 1..m}: x[i, k] + sM[i, k] - sP[i, k] = p[i] - t[i, k] + MM * yy[i, k];
subject to C4 {i in 1..n, k in 1..m}: z[i, k] ≤ MM * (1 - yy[i, k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z[i, k] = 1;
subject to C6 : sum{i in 1..n, k in 1..m} z[i, k] = n;
subject to C7 {k in 1..m}: x[1, k] - x[2, k] ≤ -t[1, k];

```

Fonte: Autoria própria

Como esperado, o modelo não utilizou de todos os colaboradores, escalonando as tarefas 1, 2 e 4 para o colaborador 1, na sequência $1 \rightarrow 2 \rightarrow 4$, devido à restrição de precedência entre as tarefas 1 e 2. E a tarefa 3 foi designada ao colaborador 3. Não houve multa para esse cenário, pois o tempo para execução de todas as tarefas não ultrapassou a janela de tempo para a realização das mesmas, que para o estudo de caso se trata de no

máximo 6 horas, 360 minutos. A Figura 14 mostra o escalonamento das tarefas, assim como o tempo de início e fim da execução de cada uma delas.

Figura 14 - Modelo para 4 tarefas e 3 colaboradores em AMPL arquivo de dados

```

----- modelo para m colaboradores-----

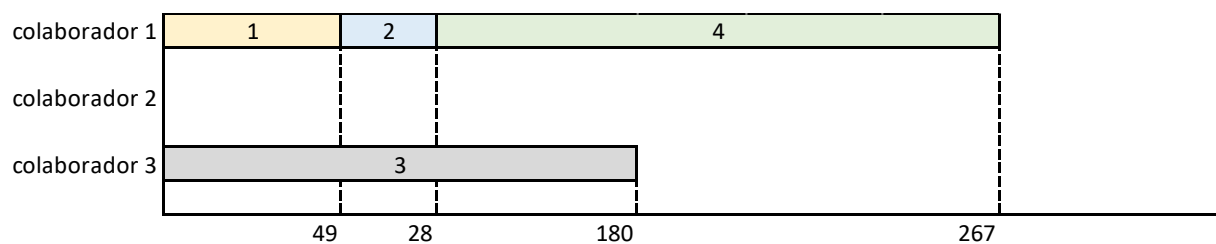
param n = 4;
param m = 3;
param MM = 100000;
param t:
    1    2    3 :=
1  49  45  56
2  28  33  40
3   -   - 180
4 190 185  -;

param p: 1    2    3    4 :=
1      60  120  200  360;
param R: 1    2    3    4 :=
1      50  40   60   30;

```

Fonte: Autoria própria

Figura 15 – Escalonamento entre 3 Colaboradores



Fonte: Autoria própria

Para determinar essa solução foram utilizadas:

- 96 variáveis, das quais 60 são binárias e 36 são lineares;
- 104 restrições ao todo, 17 de igualdade e 87 de desigualdade;
- 65 iterações simplex;
- 1 nó *branch-and-bound*.

Após análise dos primeiros resultados obtidos com o modelo, e observando que o objetivo de redução do tempo de execução através da multa aplicada foi alcançado, pode-se aplicar o mesmo, acrescentando as restrições necessárias conforme dados fornecidos, para um arquivo de dados contendo maiores informações, até alcançarmos máximo de dados do estudo de caso.

Na próxima seção será considerado um arquivo de dados para $m = 12$ colaboradores e $n = 12$ tarefas, para igualar a quantidade de tarefas e colaboradores, e certificar a minimização da multa.

4.3.2 Modelo Algébrico considerando 12 tarefas e 12 colaboradores

Utilizando da mesma declaração apresentada na seção anterior, foi solucionado o modelo em AMPL para 12 tarefas entre 12 colaboradores. Os dados serão apresentados na Tabela 2 a seguir.

Tabela 2 - Dados para escalonamento de 12 tarefas entre 12 colaboradores

Tar.	Prazo de execução (minutos)	Tempo de Execução (minutos)												Multa
		1	2	3	4	5	6	7	8	9	10	11	12	
1	60	45	46	47	51	52	53	54	55	56	57	61	62	50
2	120	32	25	39	23	24	25	26	27	28	29	33	34	40
3	360	180	181	182	240	241	-	-	244	245	246	250	251	20
4	360	257	258	259	260	264	265	248	249	250	254	255	256	30
5	120	252	253	244	255	245	246	246	251	256	257	261	262	30
6	210	-	-	-	-	-	-	-	215	-	-	-	-	50
7	240	73	61	62	66	67	68	69	70	71	72	76	77	10
8	120	-	-	-	-	180	165	-	-	-	-	-	-	60
9	160	187	188	189	186	191	192	196	190	180	181	182	197	30
10	180	-	-	-	-	125	133	-	-	-	-	-	-	60
11	200	-	-	-	-	-	-	-	240	230	-	-	-	50
12	210	-	-	-	-	-	-	45	-	-	-	-	-	60

Fonte: Autoria própria

Na da Tabela 2 é possível observar que a Tarefa 3 não pode ser realizada pelos colaboradores 6 e 7. A Tarefa 6 pode ser realizada apenas pelo colaborador 8. As Tarefas 8 e 10 podem ser realizadas apenas pelos colaboradores 5 e 6. A Tarefa 11 pode ser realizada apenas pelos colaboradores 8 e 9. E a Tarefa 12 pode ser executada apenas pelo colaborador 7.

A Figura 15 mostra que além da precedência de tarefas no modelo anterior, houve o acréscimo da restrição de precedência da Tarefa 6 e a Tarefa 10.

Figura 16 - Modelo para 12 tarefas e 12 colaboradores em AMPL

```

----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n,k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n,k in 1..m} ≥ 0;
var y{i in 1..n,j in 1..n,k in 1..m} binary;
var yy{i in 1..n,j in 1..m} binary;
var z{i in 1..n,k in 1..m} binary;
var o{i in 1..n,k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{i in 1..n} R[i] * sP[i];
subject to C1 {i in 1..n,j in 1..n,k in 1..m : i != j}: MM * y[i,j,k] + (x[i,k] - x[j,k]) ≥ t[j,k];
subject to C2 {i in 1..n,j in 1..n,k in 1..m : i != j}: MM * (1 - y[i,j,k]) + (x[j,k] - x[i,k]) ≥ t[i,k];
subject to C3 {i in 1..n,k in 1..m}: x[i,k] + sM[i,k] - sP[i,k] = p[i] - t[i,k] + MM * yy[i,k];
subject to C4 {i in 1..n,k in 1..m}: z[i,k] ≤ MM * (1 - yy[i,k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z[i,k] = 1;
subject to C6 : sum{i in 1..n,k in 1..m} z[i,k] = n;
subject to C7 {k in 1..m}: x[1,k] - x[2,k] ≤ -t[1,k];
subject to C8 {k in 1..m}: x[6,k] - x[10,k] ≤ -t[6,k];

```

Fonte: Autoria própria

Figura 17 - Modelo para 12 tarefas e 12 colaboradores em AMPL arquivo de dados

----- modelo para m colaboradores-----													
<i>param n = 12;</i>													
<i>param m = 12;</i>													
<i>param MM = 100000;</i>													
<i>param t:</i>													
	1	2	3	4	5	6	7	8	9	10	11	12	:=
1	45	46	47	51	52		53	54	55	56	57	61	62
2	32	25	39	23	24		25	26	27	28	29	33	34
3	180	181	182	240	241		999	999	244	245	246	250	251
4	257	258	259	260	264		265	248	249	250	254	255	256
5	252	253	244	255	245		246	247	251	256	257	261	262
6	999	999	999	999	999		999	999	215	999	999	999	999
7	73	61	62	66	67		68	69	70	71	72	76	77
8	999	999	999	999	999	180	165	999	999	999	999	999	999
9	187	188	189	186	191	192	196	190	180	181		182	197
10	999	999	999	999	999	125	133	999	999	999		999	999
11	999	999	999	999	999	999	999	999	240	230	221	222	
12	999	999	999	999	999	999	45	999	999	999	999	999	999;
<i>param p:</i>													
1		60	120	360	360	120		210	240	120	160	180	220 210;
<i>param R:</i>													
1		50	40	20	30	30	50		10	60	30	60	50 60;

Fonte: Autoria própria

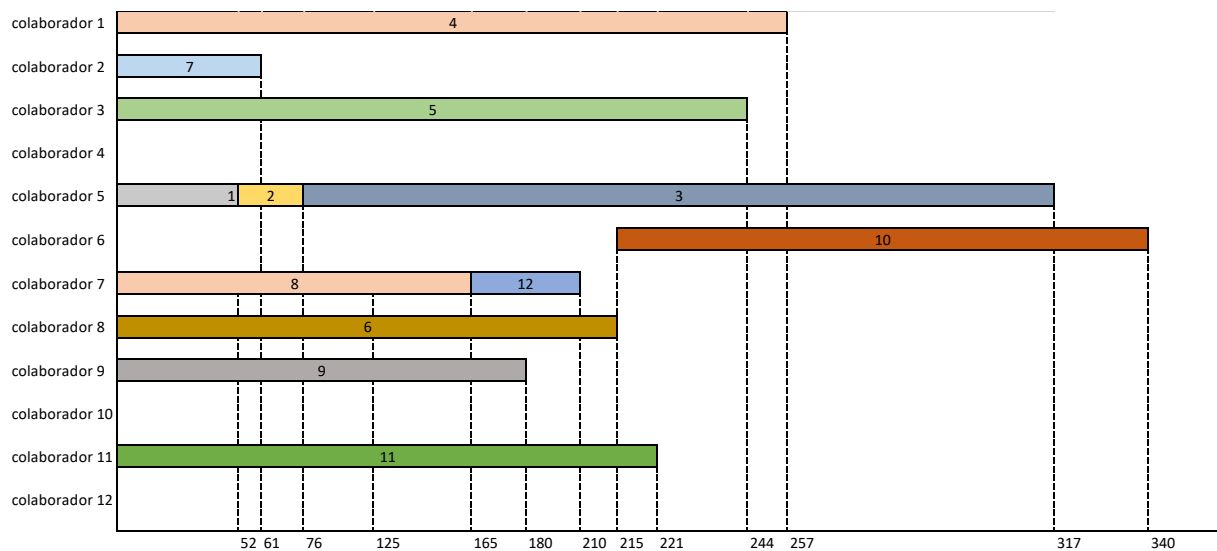
Novamente o modelo otimizou a quantidade de colaboradores envolvidos para realização e todas as tarefas.

Na Figura 18 podemos observar o escalonamento das 12 tarefas entre 9 colaboradores. Assim como o tempo de início e fim de cada tarefas, cumprindo a restrição de precedência entre elas.

Para determinar essa solução foram utilizadas:

- 2304 variáveis, das quais 1872 são binárias e 432 são lineares;
- 3493 restrições ao todo, 157 de igualdade e 3336 de desigualdade;
- 260 iterações simplex;
- 1 nó *branch-and-bound*.

Figura 18 - Escalonamento entre colaboradores



Fonte: Autoria própria.

Nesse caso houve a redução, porém uma multa foi gerada, pelo fato de as Tarefas 6, 8, 9, 10, 11 e 12 não terem sido realizadas dentro do prazo de execução estabelecido.

Na próxima seção será considerado um arquivo de dados para $m = 33$ colaboradores e $n = 35$ tarefas, máximo encontrado para solução computacional que atendesse os interesses do estudo de caso em questão.

4.4.3 Modelo Algébrico considerando 30 tarefas e 28 colaboradores

Nessa seção será apresentado o modelo com o número máximo de dados de entrada suportado pelo *Solver*. A partir desse número o *Software* retornava a seguinte mensagem: “Seu trabalho foi encerrado porque excedeu a memória máxima alocada para um trabalho.”

Figura 19 – Modelo para 33 tarefas e 30 colaboradores em AMPL

```

----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n, k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n, k in 1..m} ≥ 0;

```

```

var y{i in 1..n,j in 1..n,k in 1..m} binary;
var yy{i in 1..n,j in 1..m} binary;
var z{i in 1..n,k in 1..m} binary;
var o{i in 1..n,k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{i in 1..n} R[i] * sP[i];
subject to C1 {i in 1..n,j in 1..n,k in 1..m : i ≠ j}: MM * y[i,j,k] + (x[i,k] - x[j,k]) ≥ t[j,k];
subject to C2 {i in 1..n,j in 1..n,k in 1..m : i ≠ j}: MM * (1 - y[i,j,k]) + (x[j,k] - x[i,k]) ≥ t[i,k];
subject to C3 {i in 1..n,k in 1..m}: x[i,k] + sM[i,k] - sP[i,k] = p[i] - t[i,k] + MM * yy[i,k];
subject to C4 {i in 1..n,k in 1..m}: z[i,k] ≤ MM * (1 - yy[i,k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z[i,k] = 1;
subject to C6 : sum{i in 1..n,k in 1..m} z[i,k] = n;
subject to C7 {k in 1..m}: x[1,k] - x[5,k] ≤ -t[1,k];
subject to C8 {k in 1..m}: x[2,k] - x[6,k] ≤ -t[2,k];
subject to C9 {k in 1..m}: x[3,k] - x[7,k] ≤ -t[3,k];
subject to C10 {k in 1..m}: x[4,k] - x[8,k] ≤ -t[4,k];
subject to C11 {k in 1..m}: x[33,k] - x[17,k] ≤ -t[33,k];
subject to C12 {k in 1..m}: x[22,k] - x[28,k] ≤ -t[22,k];

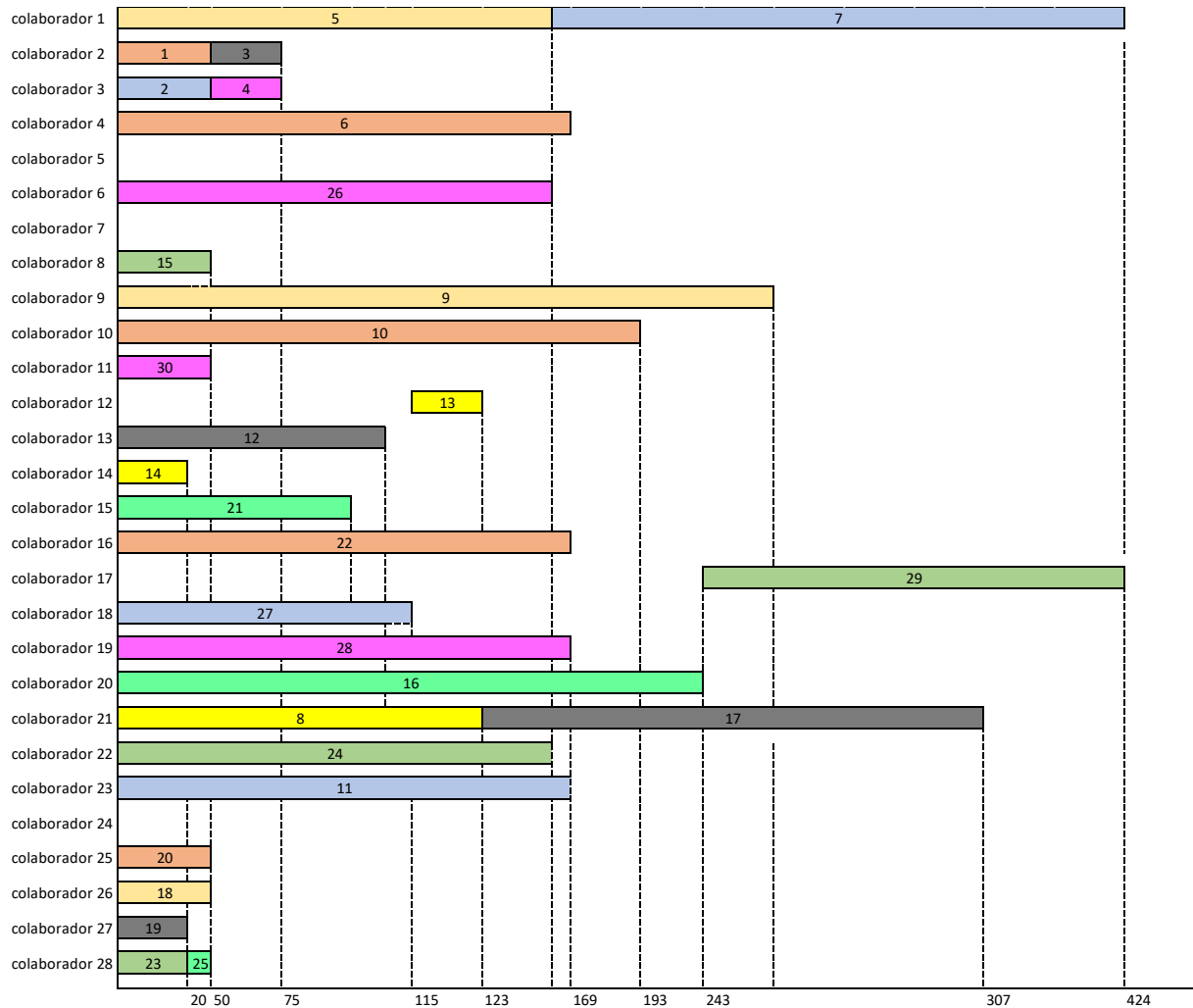
```

Fonte: Autoria própria

O modelo em AMPL para arquivo de dados consta no APÊNDICE A.

Com o máximo de memória alocada para o trabalho, o modelo otimizou a quantidade de colaboradores envolvidos para realização e todas as tarefas. Na Figura 20 podemos observar o escalonamento das 30 tarefas entre 28 colaboradores. Assim como o tempo de início e fim de cada tarefas, cumprindo a restrição de precedência entre elas.

Figura 20 - Escalonamento entre 28 colaboradores



Fonte: Autoria própria

Para determinar essa solução foram utilizadas:

- 28560 variáveis, das quais 26040 são binárias e 2520 são lineares;
- 50543 restrições ao todo, 871 de igualdade e 49672 de desigualdade;
- 2386422 iterações simplex;
- 38272 nó *branch-and-bound*.

Na seção a seguir será proposto o modelo para redução do número de colaboradores, ao invés da redução da multa.

4.4 SOLUÇÃO PROPOSTA PARA REDUÇÃO DE COLABORADORES

Para possibilitar a análise comparativa realizada posteriormente, aplicou-se o modelo de redução da quantidade de colaboradores nos mesmos três cenários apresentados na seção anterior.

4.4.1 Modelo Algébrico considerando 4 tarefas e 3 colaboradores

Tendo em vista uma análise comparativa coesa, propõe-se um modelo que busque otimizar a quantidade de colaboradores, utilizando os mesmos dados de entrada apresentados na Tabela 1.

Utilizando das mesmas especificidades entre as tarefas para cada restrição imposta no problema, o modelo em AMPL a ser resolvido utilizando o Solver GUROBI, pode ser observado na Figura 21, e o arquivo de dados pode ser visto na Figura 14, que consta na seção anterior.

Figura 21 - Modelo para 4 tarefas e 3 colaboradores em AMPL

```

----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n, k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n, k in 1..m} ≥ 0;
var y{i in 1..n, j in 1..n, k in 1..m} binary;
var yy{i in 1..n, j in 1..m} binary;
var z{i in 1..n, k in 1..m} binary;
var op{i in 1..n, k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{k in 1..m} op[k];
subject to C1 {i in 1..n, j in 1..n, k in 1..m : i ≠ j}: MM * y[i, j, k] + (x[i, k] - x[j, k]) ≥ t[j, k];
subject to C2 {i in 1..n, j in 1..n, k in 1..m : i ≠ j}: MM * (1 - y[i, j, k]) + (x[j, k] - x[i, k]) ≥ t[i, k];
subject to C3 {i in 1..n, k in 1..m}: x[i, k] + sM[i, k] - sP[i, k] = p[i] - t[i, k] + MM * yy[i, k];
subject to C4 {i in 1..n, k in 1..m}: z[i, k] ≤ MM * (1 - yy[i, k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z[i, k] = 1;
subject to C6 : sum{i in 1..n, k in 1..m} z[i, k] = n;

```

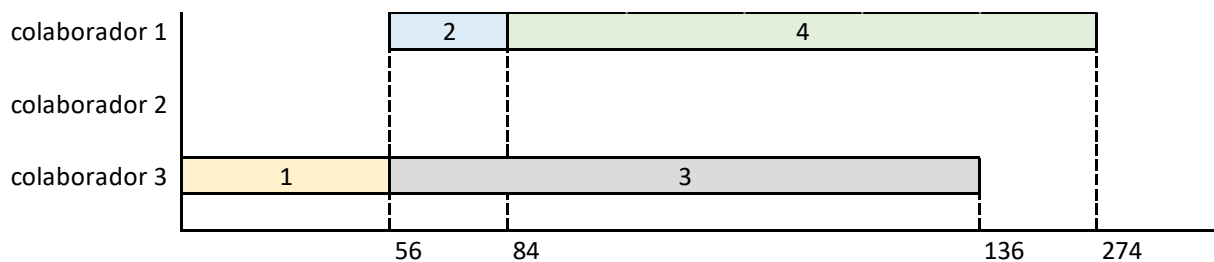

$subject\ to\ C7\ \{i\ in\ 1..n, k\ in\ 1..m\}: op[k] \geq z[i, k];$
 $subject\ to\ C8\ \{k\ in\ 1..m\}: op[k] \leq sum\{i\ in\ 1..n\} z[i, k];$
 $subject\ to\ C9\ \{k\ in\ 1..m\}: sum\{i\ in\ 1..n\} z[i, k] * t[i, k] \leq 360;$

Fonte: Autoria própria

Como esperado, o modelo não utilizou de todos os colaboradores, escalonando as tarefas 2 e 4 para o colaborador 1, na sequência $2 \rightarrow 4$, devido à restrição de precedência entre as tarefas 1 e 2, que foi considerada ainda nesse caso. E as tarefas 1 e 3 foram designadas ao colaborador 3. O tempo para execução de todas as tarefas não ultrapassou a janela de tempo para a realização das mesmas, que para o estudo de caso se trata de no máximo 6 horas, 360 minutos.

A Figura 22 mostra o escalonamento das tarefas, assim como o tempo de início e fim da execução de cada uma delas.

Figura 22 - Designação de 4 tarefas



Fonte: Autoria própria

Para determinar essa solução foram utilizadas:

- 93 variáveis, das quais 57 são binárias e 36 são lineares;
- 107 restrições ao todo, 16 de igualdade e 91 de desigualdade;
- 13 iterações simplex.

Após análise dos primeiros resultados obtidos com o modelo, e observando que o objetivo de redução na quantidade de colaboradores envolvidos no projeto foi alcançado, pode-se aplicar o mesmo, acrescentando as restrições necessárias conforme dados fornecidos, para um arquivo de dados contendo as mesmas informações do modelo apresentado anteriormente.

Na próxima seção será considerado um arquivo de dados para $m = 12$ colaboradores e $n = 12$ tarefas, para igualar a quantidade de tarefas e colaboradores, e certificar a minimização do número de colaboradores.

4.4.2 Modelo Algébrico considerando 12 tarefas e 12 colaboradores

Utilizando das mesmas especificidades entre as tarefas para cada restrição imposta no problema, o modelo em AMPL a ser resolvido utilizando o Solver GUROBI, pode ser observado na Figura 23, e o arquivo de dados pode ser visto na Figura 16, que consta na seção anterior.

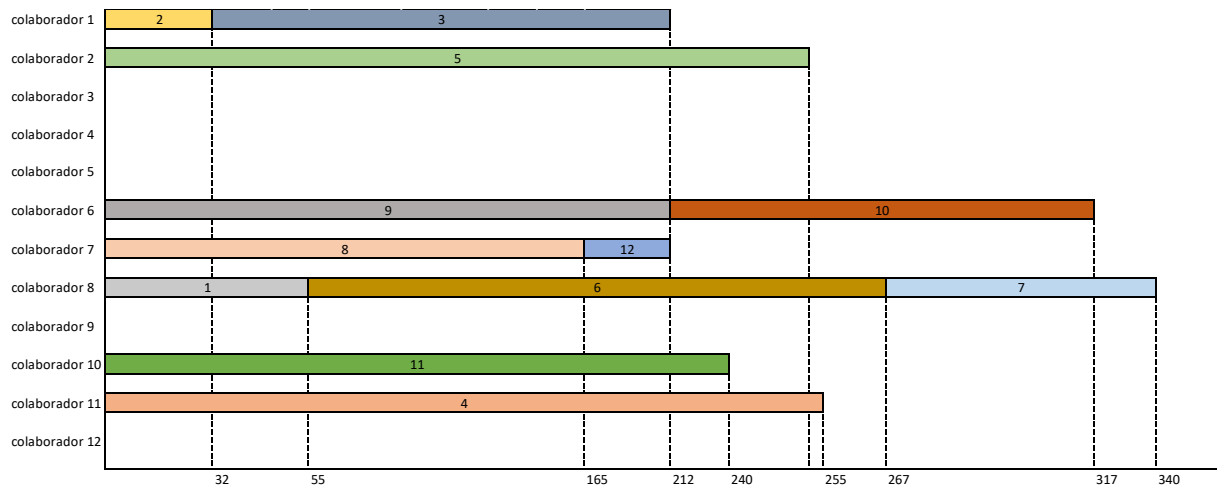
Figura 23 - Modelo para 12 tarefas e 12 colaboradores em AMPL

```
----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n, k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n, k in 1..m} ≥ 0;
var y{i in 1..n, j in 1..n, k in 1..m} binary;
var yy{i in 1..n, j in 1..m} binary;
var z{i in 1..n, k in 1..m} binary;
var op{i in 1..n, k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{k in 1..m} op[k];
subject to C1 {i in 1..n, j in 1..n, k in 1..m : i != j}: MM * y[i, j, k] + (x[i, k] - x[j, k]) ≥ t[j, k];
subject to C2 {i in 1..n, j in 1..n, k in 1..m : i != j}: MM * (1 - y[i, j, k]) + (x[j, k] - x[i, k]) ≥ t[i, k];
subject to C3 {i in 1..n, k in 1..m}: x[i, k] + sM[i, k] - sP[i, k] = p[i] - t[i, k] + MM * yy[i, k];
subject to C4 {i in 1..n, k in 1..m}: z[i, k] ≤ MM * (1 - yy[i, k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z[i, k] = 1;
subject to C6 : sum{i in 1..n, k in 1..m} z [i, k] = n;
subject to C7 {i in 1..n, k in 1..m}: op[k] ≥ z [i, k];
subject to C8 {k in 1..m} : op[k] ≤ sum{i in 1..n} z [i, k];
subject to C9 {k in 1..m}: sum{i in 1..n} z [i, k] * t[i, k] ≤ 360;
```

Fonte: Autoria própria

Novamente o modelo otimizou a quantidade de colaboradores envolvidos para realização e todas as tarefas. Na Figura 24 podemos observar o escalonamento das 12 tarefas entre 7 colaboradores. Assim como o tempo de início e fim de cada tarefas, sem nenhuma restrição de precedência entre elas.

Figura 24 - Designação de 12 tarefas



Fonte: Autoria própria

Para determinar essa solução foram utilizadas:

- 2254 variáveis, das quais 1822 são binárias e 432 são lineares;
- 3506 restrições ao todo, 155 de igualdade e 3351 de desigualdade;
- 473 iterações simplex.

Nesse caso, houve a percepção de que a restrição de precedência de tarefa conflitou com a restrição da soma dos tempos de execução das tarefas por um mesmo colaborador. Onde, com a retirada da restrição de precedência de tarefas alcançou-se o objetivo de minimizar a quantidade de colaboradores em 7 colaboradores para executar as 12 tarefas.

Na próxima seção será considerado um arquivo de dados para $m = 28$ colaboradores e $n = 30$ tarefas, máximo encontrado para solução computacional que atendesse os interesses do estudo de caso em questão.

4.4.3 Modelo Algébrico considerando 30 tarefas e 28 colaboradores

Utilizando das mesmas especificidades entre as tarefas para cada restrição imposta no problema, o modelo em AMPL a ser resolvido utilizando o Solver GUROBI, pode ser observado na Figura 25, e o arquivo de dados pode ser visto no APÊNDICE A.

Figura 25 - Modelo para 30 tarefas e 28 colaboradores em AMPL

```

----- modelo para m colaboradores-----
param n;
param m;
param MM;
param t{i in 1..n,k in 1..m};
param p{i in 1..n};
param R{i in 1..n};
var x{i in 1..n,k in 1..m} ≥ 0;
var y{i in 1..n,j in 1..n,k in 1..m} binary;
var yy{i in 1..n,j in 1..m} binary;
var z{i in 1..n,k in 1..m} binary;
var op{i in 1..n,k in 1..m} binary;
var sM{i in 1..n} ≥ 0;
var sP{i in 1..n} ≥ 0;
Minimize FO: sum{k in 1..m} op[k];
subject to C1 {i in 1..n,j in 1..n,k in 1..m : i != j}: MM * y[i,j,k] + (x[i,k] - x[j,k]) ≥ t[j,k];
subject to C2 {i in 1..n,j in 1..n,k in 1..m : i != j}: MM * (1 - y[i,j,k]) + (x[j,k] - x[i,k]) ≥ t[i,k];
subject to C3 {i in 1..n,k in 1..m}: x[i,k] + sM[i,k] - sP[i,k] = p[i] - t[i,k] + MM * yy[i,k];
subject to C4 {i in 1..n,k in 1..m}: z[i,k] ≤ MM * (1 - yy[i,k]);
subject to C5 {i in 1..n}: sum {k in 1..m} z[i,k] = 1;
subject to C6 : sum{i in 1..n,k in 1..m} z[i,k] = n;
subject to C7 {i in 1..n,k in 1..m}: op[k] ≥ z[i,k];
subject to C8 {k in 1..m} : op[k] ≤ sum{i in 1..n} z[i,k];
subject to C9 {k in 1..m}: sum{i in 1..n} z[i,k] * t[i,k] ≤ 360.

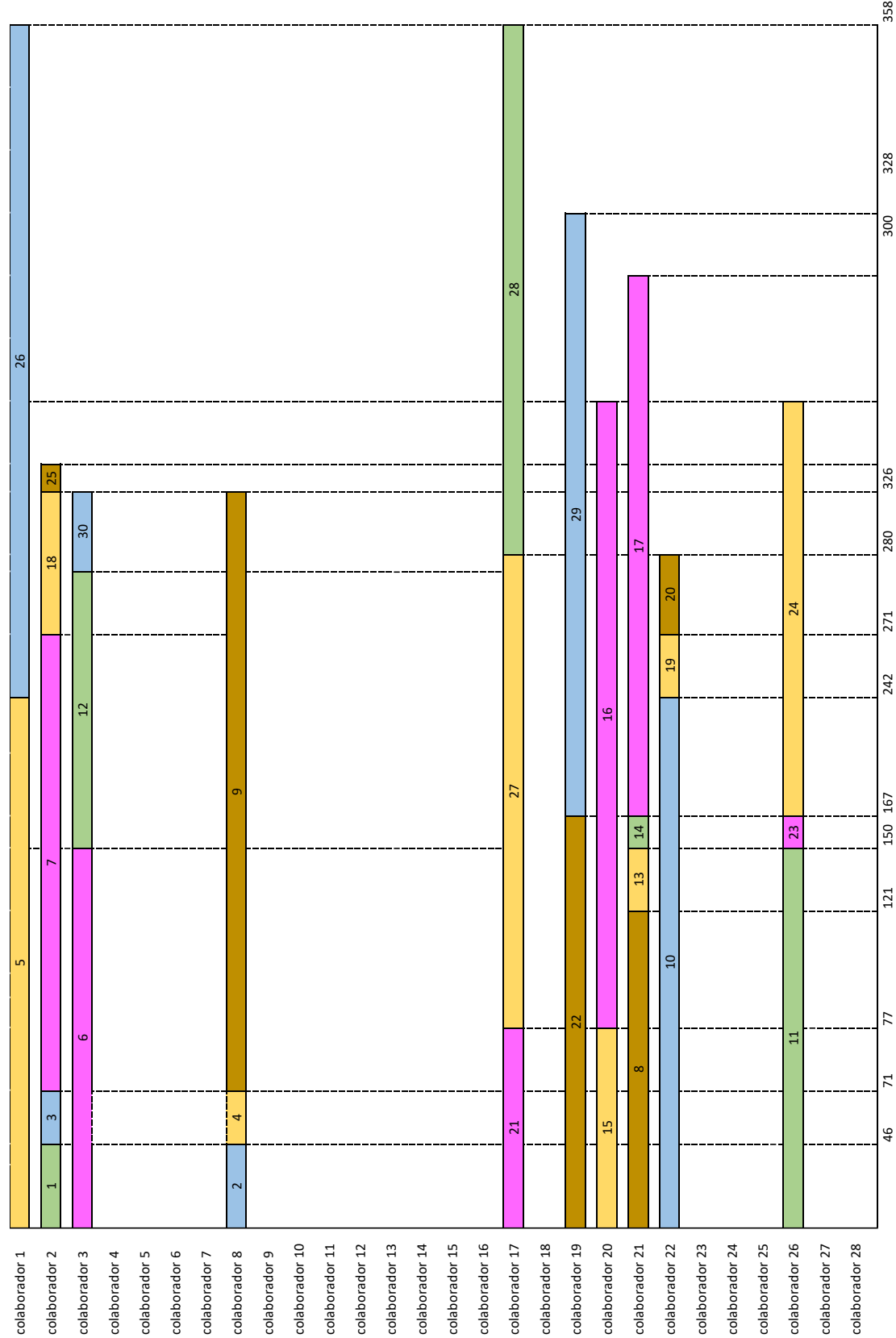
```

Fonte: Autoria própria

Novamente o modelo otimizou a quantidade de colaboradores envolvidos para realização e todas as tarefas.

Na Figura 26 podemos observar o escalonamento das 30 tarefas entre 10 colaboradores.

Figura 26 – Designação de 30 tarefas



Fonte: Autoria própria

Para determinar essa solução foram utilizadas:

- 28321 variáveis, das quais 25801 são binárias e 2502 são lineares;
- 50793 restrições ao todo, 871 de igualdade e 49922 de desigualdade;
- 3064 iterações simplex;

4.5 ANÁLISE DOS MÉTODOS DE SOLUÇÃO

Nesta etapa será realizada a análise dos modelos de otimização, após a apresentação dos resultados obtidos através dos dois métodos de resolução considerados: modelo exato para redução da multa e modelo exato para redução de colaboradores.

Ambos os modelos apresentaram soluções ótimas, que minimizaram o número de colaboradores envolvidos no projeto.

Para o modelo que possuía como função-objetivo minimizar a multa por atraso na execução das tarefas, se encontra benefício devido a restrição de precedência de tarefas ser considerada para o tempo de início de cada tarefa. Porém, em comparação com o segundo modelo apresentado, o número de colaboradores resultante encontrado foi superior.

Por se tratar de um modelo mais complexo, o tempo de resolução é consideravelmente maior, onde o *software* limitou o número de entrada de dados devido sua capacidade de alocação de memória.

No modelo de redução do número de colaboradores, o resultado final pode ser avaliado como melhor nos três cenários realizados, levando em conta apenas questões quantitativas. Pois para a análise do problema como um todo o uso desse modelo se torna obsoleto, devido o tempo de início das tarefas não seguir a aplicação de ordem de prioridade, e por consequência sua restrição de precedência também não ser efetiva.

Em termos de complexidade e tempo computacional, é um modelo mais simples.

5 CONSIDERAÇÕES FINAIS

Os problemas de escalonamento de situações reais são resolvíveis na prática por pessoas qualificadas e experientes, mas que dependendo da sua competência, podem influenciar diretamente no planejamento final de um projeto. Ressalta-se que qualquer situação cotidiana é passível de otimização, uma vez que possui algumas variáveis de decisão.

Para o desenvolvimento deste trabalho, foi observado numa Associação que promove eventos de cunho religiosos, onde observou-se em um dos eventos realizados recursos humanos dispensáveis para a execução das tarefas envolvidas na montagem do mesmo. Durante o desenvolvimento do modelo proposto percebeu-se a importância da aplicação em cenários com número de tarefas e colaboradores diferentes para validação do modelo, garantindo, em um primeiro passo, a resposta esperada através da aplicação com um número reduzido de tarefas e colaboradores.

O objetivo principal desse trabalho foi minimizar o custo relativo a quantidade de colaboradores envolvidos no projeto real que dispunha de tarefas de preparação, organização e montagem do evento, as quais dispunham de um período de 6 horas no total para serem realizadas, porém cada uma com duração e prazos específicos. De posse desses valores, foram calculados os tempos necessários para execução de cada tarefa por cada colaborador, tendo em vista que, por tratar-se de recursos humanos, a empresa forneceu médias relativas ao tempo de execução de cada tarefa, desde o colaborador mais competente, até o mais inexperiente, para assim, diminuir custos desnecessários quando utilizados.

Para obtenção de resultados foi proposto à utilização do método exato, através da solução do modelo matemático escrito em AMPL, utilizando o *solver* GUROBI, disponível para problemas de programação linear inteira e mista.

A grande divergência encontrada na comparação dos modelos está relacionada ao objetivo do problema proposto, porquanto que a precedência entre as tarefas especificadas não pode ser inseridas no modelo de redução de colaboradores, para gerar um procedimento de melhoria aproximado do caso real. Ou seja, na prática alguns fatores influenciam para a aplicação do resultado, como por exemplo os relacionados a interrupções não previstas durante à execução da tarefa de um colaborador, sendo este um fator limitante da implementação do método ao projeto no caso real.

Ainda assim, em relação aos resultados obtidos, considerou-se inviável continuar com a prática utilizada pela Associação atualmente, devido ao alto custo com mão de obra e

processos de designação de tarefas não eficientes. Isto porque a redução encontrada na otimização dos recursos resultou em uma economia no recrutamento de colaboradores para a organização, propondo para o projeto do próximo evento a ser promovido uma redução de até 64% no número de colaboradores para a execução das tarefas. Dependendo da abordagem ao modelo matemático de otimização de escalonamento de tarefas entre os colaboradores a ser utilizado.

5.1 TRABALHOS FUTUROS

Para trabalhos futuros sugere-se o estudo das propriedades estruturais e a implementação de problemas de programação da produção em outros ambientes, como por exemplo, operadores no modelo *job shop*, incluindo restrições diferentes das tratadas nesta pesquisa, como *setup* independente da sequência, janelas de entrega e de minimização simultânea do tempo ocioso e o tempo de execução das tarefas, com janela de tempo de execução. Outra direção interessante para trabalhos futuros, seria o aperfeiçoamento do modelo de redução de colaboradores, para considerar o tempo do início das tarefas e suas restrições de precedência.

REFERÊNCIAS

- ANDRADE, L. **Introdução à pesquisa operacional: métodos e modelos para análise de decisões**. 4 ed. Rio de Janeiro: LTC, 2014.
- ARENALES, Marcos, ARMENTANO, Vinicius, MORABITO, Reinaldo, YANASSE, Horácio. **Pesquisa operacional**. Rio de Janeiro, 2007.
- ARNOLD, J. R. **Introduction to Materials Management**. Columbus, Ohio: Prentice Hall, 1998. 521 p.
- BAKER, K. R. **A dynamic priority rule for scheduling against due-dates**. JWM Bertrand. Journal of Operations Management, 1982. 37-42 p.
- BAKER, K. R. **Introduction to Sequencing and Scheduling**. John Wiley. New York, 1974. 112 p.
- BALLOU, R. H. **Gerenciamento da cadeia de suprimentos/logística empresarial**. 5 ed. Porto Alegre: Bookman, 2006. 616p
- BARBOSA, L. B.. **Otimização do Sequenciamento de Tarefas em Máquinas Paralelas com Tempos de Processamentos Diferentes**. (Graduação), Universidade Federal de Santa Catarina, 2017.
- BARROS, J. R. F. e TUBINO, D. F. **O Planejamento e Controle da Produção nas Pequenas Empresas – Uma Metodologia de Implantação**. 1998. Disponível em <http://www.abepro.org.br/biblioteca/ENEGEP1998_ART262.pdf>. Acesso em agosto de 2018.
- BLAZEWICZ, J., ECKER, K. H., PESCH, E., SCHMIDT, G., WEGLARZ, J. Scheduling computer and manufacturing processes. In **Springer Verlag**, 1996.
- BLAZEWICZ, J. et al. **Handbook on planejamento: from theory to applications**. Heidelberg: Springer Verlag, 2007.
- BOWERSOX, D. J.; CLOSS, D. J. **Logística empresarial: o processo de integração da cadeia de suprimento**. São Paulo: Atlas, 2011. 594 p.
- BRUCKER P, DREXL A, MOHRING R, NEUMANN K, PESCH E, *Resource-constrained project scheduling: notation, classification, models, and methods*. European Journal of Operational Research 112: 3-41, 1999.

BRUCKER, Peter et al. *A branch and bound algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research.* v. 107, n. 2, p. 272–288, 1998. ISSN 0377-2217. Disponível em:

<<http://www.sciencedirect.com/science/article/B6VCT3VGMD1N-4/2/4a3052fdd1d9541d8b94aecb007b3c8b>>. Acesso em agosto de 2018.

COLIN, E. C. **Pesquisa Operacional: 170 aplicações em Estratégia, Finanças, Logística, Produção, Marketing e vendas.** Rio de Janeiro: LTC, 2013. 501 p.

CRAVO, Gildásio Lecchi. **Escalonamento de projetos com restrições de recursos e múltiplos modos de processamento: soluções heurísticas e uma aplicação à programação de manutenção industrial.** (Tese), Universidade Federal do Espírito Santo, 2009.

FIDELIS, M. B.; ARROYO, J. E. C. **Heurísticas para o problema de sequenciamento de lotes de tarefas em máquinas paralelas.** XLIX Simpósio Brasileiro de Pesquisa Operacional. Blumenau, 2017.

GIGANTE, Rodrigo Luiz. **Heurística construtiva para a programação de operações flow shop permutacional.** Dissertação (Mestrado). 90 p. 2010. Escola de Engenharia de São Carlos da Universidade de São Paulo. São Paulo, 2010.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização Combinatória e Programação Linear: modelos e algoritmos.** 2. ed. Rio de Janeiro: Elsevier, 2005.

GRAVES, S. C. *A Review of Production Scheduling. Operations Research*, v29, p. 646-675, 1981.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**, 9 ed. Porto Alegre: AMGH, 2013. 1005 p.

HOCHBAUM, D. S. **The Scheduling Problem.** RIOT: Remote Interactive Optimization Testbed, 1999. Disponível em: <<http://riot.ieor.berkeley.edu/Applications/Scheduling/algorithms.html>>. Acesso em: outubro 2018.

LEAL, A. J. S. **Algoritmos de investigação operacional para um problema de sequenciamento de projetos.** (Mestrado), Universidade de Minho, Portugal, 2007.

LE PAPE, C. **Constraint-based scheduling: A tutorial.** Proceedings of the 1st International Summer School on Constraint Programming, 2005.

MONACI, M.; FAÊDA, F. M.; ARROYO, J. E. C.; SANTOS, A. G. **Sequenciamento de tarefas em máquinas paralelas com tempos de preparação e precedência entre as tarefas: modelagem e heurísticas construtivas.** XLVII SBPO. Pernambuco, 2015.

- MOREIRA, D. **Administração da produção e operações**. 2. ed. São Paulo: Pioneira, 1986.
- NEOS GUIDE SERVER, User's Guide to the NEOS Server, Disponível em <<https://neos-guide.org/content/users-guide>>. Acesso em Setembro de 2018.
- NAGANO, M.S. **Evaluating the performance of tabu search procedures for flow shop sequencing**. Journal of the Operational Research Society 49 (12), 1296-1302, 1998.
- NEGREIRO, Marcos; BARBOSA, Willame Tiberio. **O problema de alocação de recursos e seleção de múltiplos projetos de ti**. Revista de Gestão e Projetos, 4(2): 27–49, mai./ago. 2013.
- NEOS GUIDE SERVER, **User's Guide to the NEOS Server**, Disponível em <<https://neos-guide.org/content/users-guide>>. Acesso em Agosto de 2018.
- OLIVEIRA, Ivone de Lourdes. **Dimensão estratégica da comunicação no contexto organizacional contemporâneo: um paradigma de interação comunicacional dialógica**. Tese (Doutorado) – UFRJ, Escola de Comunicação, 2002.
- PINEDO, M. L. **Planning and planejamento in manufacturing and services**. Springer Series Operation Research: New York, USA, 2005.
- PINEDO, M. **Scheduling: theory, algorithms and systems**. Prentice Hall, New Jersey. 2008.
- PINEDO, M. **Scheduling: theory, algorithms, and systems**. Englewood Cliffs, NJ, 1995.
- PÉREZ, Miguel Angel Fernández. **Um método heurístico para o problema de escalonamento multiobjetivo em vários ambientes de máquinas**. (Dissertação de Mestrado). Rio de Janeiro, 2012.
- PMI. *A Guide To The Project Management Body Of Knowledge* (PMBOK Guides). [S.l.]: Project Management Institute, 2008.
- TAHA, H. A. **Pesquisa Operacional**. 8. ed. São Paulo: Pearson, 2008. 359 p.
- TAVARES, L. *A review of the contribution of operational research to project management*, European Journal of Operational Research 136: 1-18, 2002.
- TCHAO, C. **Heurísticas para o Problema de Escalonamento de Projetos com Restrição de Recursos**. Dissertação (Mestrado) — Dissertação de Mestrado em Ciência da Computação da Universidade Federal Fluminense, 2007.

TRICHES, J.; KRIPKA, M.; BOSCARDIN, I. **Otimização aplicada ao problema de alocação de equipes em uma panificadora**. *Exacta – EP*, São Paulo, v. 13, n. 3, p. 377-388, 2015.

TUBINO, Dalvio Ferrari. **Manual de Planejamento e Controle da Produção**. 2. ed. São Paulo: Atlas, 2007. 215 p.

SANTO. L. C. DO E. **Sequenciamento de máquinas através de algoritmos genéticos**. Trabalho de Conclusão de Graduação, Londrina, 2013. Disponível em: <<http://www.uel.br/cce/dc/wp-content/uploads/VersaoPreliminarTCC-LauraEspiritoSanto.pdf>>. Acesso em Setembro de 2018.

SILVA, André Renato Villela da. **Um método híbrido para um problema de escalonamento de projetos**. In **Congresso Latino-Iberoamericano de Investigación Operativa**. Simpósio Brasileiro de Pesquisa Operacional, Setembro 2012.

SOUZA, Eduardo Cordeiro. **Programação de tarefas em um *flow shop***. Tese (Doutorado). 124 p. 2009. Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Naval e Oceânica. São Paulo, 2009.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da produção**. 2. ed. São Paulo: Atlas, 2002.

FLICKR, Onda Dura. Disponível em: <<https://www.flickr.com/photos/ondaduraoficial/albums/72157693053419412/with/27883112718/>> Acesso em: 05 de novembro de 2018.

APÊNDICE A – DADOS

param n=30;																												
param m=28;																												
param MM=100000;																												
param t:																												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28:=
1	45	46	48	49	50	51	52	53	56	58	59	60	63	64	1000	1000	1000	1000	1000	98	99	107	108	109	110	111	114	115
2	46	47	49	50	51	52	53	54	57	59	60	61	64	65	1000	1000	1000	1000	1000	99	100	108	109	110	111	112	115	116
3	32	25	20	21	22	23	24	25	28	30	31	32	35	36	1000	1000	1000	1000	1000	70	71	79	80	81	82	83	86	87
4	33	26	21	22	23	24	25	26	29	31	32	33	36	37	1000	1000	1000	1000	1000	71	72	80	81	82	83	84	87	88
5	180	181	183	184	185	240	241	242	245	247	248	249	252	253	1000	1000	1000	1000	1000	152	153	205	206	207	208	209	212	213
6	181	182	184	185	186	241	242	243	246	248	249	250	253	254	1000	1000	1000	1000	1000	153	154	206	207	208	209	210	213	214
7	182	183	185	186	187	242	242	242	242	242	242	242	242	242	1000	1000	1000	1000	1000	122	122	202	202	202	202	202	202	202
8	183	184	186	187	188	243	243	243	243	243	243	243	243	243	1000	1000	1000	1000	1000	123	123	203	203	203	203	203	203	203
9	248	249	251	252	253	254	255	256	259	261	262	263	266	267	1000	1000	1000	1000	1000	301	302	310	311	312	313	314	317	318
10	180	181	183	184	185	186	187	188	191	193	194	195	198	199	1000	1000	1000	1000	1000	233	234	242	243	244	245	246	249	250
11	245	246	248	249	250	251	252	253	256	258	259	260	263	264	1000	1000	1000	1000	1000	153	154	162	163	164	165	166	169	170
12	80	81	83	84	85	86	87	88	91	93	94	95	98	99	1000	1000	1000	1000	1000	133	134	142	143	144	145	146	149	150
13	20	21	23	24	25	26	27	28	16	18	19	20	23	24	45	47	18	19	20	28	29	20	21	22	23	24	27	28
14	12	13	15	16	17	18	19	20	23	10	11	12	15	16	22	24	25	26	27	16	17	25	26	27	28	29	32	10
15	45	46	48	49	50	51	52	53	56	58	59	60	63	64	62	64	65	66	67	75	76	84	85	86	87	88	91	92
16	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	243	244	1000	1000	1000	1000	1000	1000	1000
17	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	183	184	1000	1000	1000	1000	1000	1000	1000
18	60	61	63	64	65	66	67	68	71	73	74	75	78	79	67	45	46	47	48	56	57	53	54	55	56	57	60	61
19	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	29	30	31	32	19	22	23
20	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	57	58	59	45	46	49	50

21	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	74	76	77	78	79	1000	1000	1000	1000	1000	1000	1000	1000	1000
22	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	180	162	163	164	165	1000	1000	1000	1000	1000	1000	1000	1000	1000
23	30	35	21	22	23	38	25	26	29	31	36	32	23	20	22	24	25	26	27	35	36	28	25	20	22	18	21	22	
24	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	152	153	154	155	156	159	160
25	10	11	13	14	15	16	17	18	21	23	24	25	28	29	29	31	32	33	34	30	28	15	17	18	25	30	12	10	
26	180	181	183	184	185	186	187	188	191	193	194	195	198	199	220	222	223	224	225	233	234	242	243	244	245	246	249	250	
27	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	111	113	114	115	116	1000	1000	1000	1000	1000	1000	1000	1000	
28	210	180	185	220	210	211	212	213	216	218	219	220	223	224	170	166	167	168	169	177	178	186	187	188	189	245	193	194	
29	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	202	180	181	182	183	1000	1000	1000	1000	1000	1000	1000	1000	
30	30	31	33	34	35	36	37	38	41	43	44	45	48	49	50	31	32	33	34	36	37	40	41	42	50	30	33	34;	
param p:=																													
1	40																												
2	40																												
3	40																												
4	40																												
5	25																												
6	25																												
7	25																												
8	25																												
9	160																												
10	160																												
11	160																												
12	160																												
13	360																												
14	240																												
15	120																												
16	120																												

17	140
18	240
19	240
20	240
21	300
22	210
23	120
24	240
25	300
26	60
27	80
28	120
29	360
30	110;

param R:=

1	50
2	50
3	50
4	50
5	40
6	40
7	40
8	40
9	20
10	20
11	20
12	20
13	30

14	40
15	30
16	30
17	20
18	20
19	20
20	20
21	20
22	50
23	20
24	10
25	10
26	10
27	60
28	60
29	50
30	40;

